

UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



NÁVRH A OPTIMALIZÁCIA ALGORITMOV NA  
ODHAĽOVANIE FRAUĐOV V DÁTACH ZDRAVOTNEJ  
POISŤOVNE

DIPLOMOVÁ PRÁCA

UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

**Návrh a optimalizácia algoritmov na odhaľovanie fraudov v  
dátach Zdravotnej poisťovne**

**DIPLOMOVÁ PRÁCA**

Študijný program: ekonomicko-finančná matematika a modelovanie  
Študijný odbor: 9.1.9 aplikovaná matematika  
Školiace pracovisko: Katedra aplikovanej matematiky a štatistiky  
Vedúci práce: Mgr. Henrieta Tulejová, MSc.



Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

---

## ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Bc. Adriána Santrová  
**Študijný program:** ekonomicko-finančná matematika a modelovanie  
(Jednoodborové štúdium, magisterský II. st., denná forma)  
**Študijný odbor:** 9.1.9. aplikovaná matematika  
**Typ záverečnej práce:** diplomová  
**Jazyk záverečnej práce:** slovenský  
**Sekundárny jazyk:** anglický

**Názov:** Návrh a optimalizácia algoritmov na odhaľovanie fraudov v dátach zdravotnej poisťovne.  
*Design and optimization of algorithms for detecting fraud in health care data.*

**Cieľ:** Cieľom práce bude navrhnúť, vyskúšať a optimalizovať rôzne algoritmy, ktoré uľahčia revíznym lekárom výber dát alebo poskytovateľov, na ktoré sa treba pri kontrole zamerať.

**Vedúci:** Mgr. Henrieta Tulejová, MSc.  
**Katedra:** FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky  
**Vedúci katedry:** prof. RNDr. Daniel Ševčovič, CSc.  
**Dátum zadania:** 21.01.2016

**Dátum schválenia:** 25.01.2016

prof. RNDr. Daniel Ševčovič, CSc.  
garant študijného programu

.....  
študent

.....  
vedúci práce

**Podakovanie** Touto cestou by som sa chcela poďakovať mojej školiteľke, pani Mgr. Henriete Tulejovej, MSc., za všetky skúsenosti, ktoré som na základe jej rozhodnutia viesť túto diplomovú prácu nadobudla. Rovnako moja vďaka patrí aj zdravotnej poisťovni Dôvera, ako za poskytnuté dáta, tak aj za možnosť vzdelávať sa u nich v oblasti zdravotníctva a zabezpečovania služieb pre poistencov. Ďalej by som sa chcela poďakovať Mgr. Marekovi Ciesarovi za mentorovanie, všetky zodpovedané otázky ohľadne poisťovne aj tejto diplomovej práce a za veľa dobrých nápadov. Taktiež chcem touto cestou prejať vďaka Fakulte matematiky, fyziky a informatiky, špeciálne Katedre aplikovanej matematiky a štatistiky, za celých 5 rokov štúdia, ktoré ma obohatili ako po intelektuálnej stránke, tak aj po tej osobnostnej, za všetky dvere, ktoré mi pomohli otvoriť a tak napredovať. Moja vďaka patrí aj mojim rodičom, za ich dlhoročnú podporu a vedenie k štúdiu, snahu pomôcť mi byť každý deň lepšia a tak aj spokojnejšia ako po profesionálnej stránke tak aj v súkromí. Nie malé ďakujem chcem vysloviť aj mojim súrodencom, ktorí ma svojou šikovnosťou a inteligenciou priamo či nepriamo motivovali k neustálemu napredovaniu, lebo kde moja vlastná motivácia končila, priateľská súťaživosť s nimi začínala a povzbudila ma k ešte lepším výsledkom. V neposlednom rade sa chcem poďakovať môjmu snúbencovi Lubomírovi, ktorý bol pri mne aj v najťažších chvíľach, podržal ma a pomohol mi ísť za svojim cieľom.

## Abstrakt v štátnom jazyku

Santrová, Adriána: Návrh a optimalizácia algoritmov na odhaľovanie fraudov v dátach zdravotnej poisťovne [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; Školiťel: Mgr. Henrieta Tulejová, MSc., Bratislava, 2017, 70str.

V našej práci sme sa zaoberali detekciou podvodov v dátach zdravotnej poisťovne. Primárnym cieľom bolo vytvoriť a neskôr zefektívniť algoritmus, ktorý by bol schopný z množstva poskytovateľov zdravotnej starostlivosti vybrať tých, ktorí sa istým spôsobom správajú neštandardne a teda stojí za zmienku podrobnejšie analyzovať ich vykázanú zdravotnú starostlivosť. Odrazili sme sa od už existujúceho jednoduchého algoritmu zdieľania poistencov dvojicami poskytovateľov. Ten sme následne rozšírili do viacrozmeru a na základe ďalších podmienok upravovali a zefektívňovali aby zvládol aj veľmi veľké sady dát. Následne sme ho použili na rôzne dátové výbery, čím sme demonštrovali jeho širokú aplikovateľnosť, teda pri malej zmene v algoritme sme boli schopní sledovať odlišné ciele v analýze.

**Kľúčové slová:** Dáta zdravotnej poisťovne, podvod, návrh algoritmu, modifikácia a aplikácia algoritmu

# Abstract

Santrová, Adriána: Design and optimization of algorithms for detecting fraud in health care data [Diploma Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: Mgr. Henrieta Tulejová, MSc., Bratislava, 2017, 70p.

Our thesis is dedicated to frauds detection in health care data. Aim was design and optimize algorithm, which would be able to select health care providers who behave in some unusual way and who should be further analyzed. We started with simple, already existing algorithm detecting patient sharing scheme, then we universalise it for more than two health care providers and optimize and modify it for large datasets. Finally, we used it on various datasets in order to demonstrate the versatility of our algorithm - a small change in algorithm allows us to monitor different behavior of health care providers.

**Keywords:** Health care data, frauds, algorithm design, modification and optimization of algorithm

# Obsah

Úvod	8
<b>1 Zdravotnícka terminológia a dáta</b>	<b>12</b>
1.1 Terminológia	12
1.2 Dáta zdravotnej poisťovne	13
1.2.1 Vykazovanie zdravotnej starostlivosti	13
1.2.2 Typy podvodov a omylov pri vykazovaní	15
<b>2 Zdieľanie poistencov</b>	<b>17</b>
2.1 Algoritmus pre dvojice poskytovateľov	18
2.2 Rozšírenie pre ľubovoľne veľké n	18
2.3 Upravený algoritmus	20
2.4 Čistenie výsledkov	26
<b>3 Aplikácia algoritmu</b>	<b>28</b>
3.1 Výber dát a spracovanie	28
3.1.1 Časové obdobie	28
3.1.2 Údaje o PZS	29
3.1.3 Údaje o POI	29
3.1.4 Údaje o VZS	30
3.2 Analýza lekární	30
3.2.1 Na celej VZS	30
3.2.2 Rok 2016	33
3.2.3 Vzdialenosť ako filter	34
3.2.4 Zdravotnícke pomôcky	37
3.3 Špeciálna ambulantná starostlivosť	38
<b>4 Podozriví poskytovatelia z analýz</b>	<b>46</b>
<b>Záver</b>	<b>47</b>
<b>Príloha</b>	<b>50</b>

## Úvod

Každý človek túži prežiť keď už nie dlhý, tak aspoň kvalitný život. Inak tomu nie je ani na Slovensku. Z posledných prieskumov [3] však vyplýva, že ako národ nie sme spokojní. Jednou z najdôležitejších stránok nášho života je aj zdravie. A práve v ňom v porovnaní s okolitými štátmi zaostávame. Kvalitu nášho zdravia meria tzv. *Index Healthy Life Years*, ktorý ukazuje počet rokov prežitých v plnom zdraví. Na Slovensku dosahuje tento ukazovateľ hodnotu 54,6 roka pre ženy a 55,5 roka pre mužov. V tomto sa nachádzame pod priemerom krajín EÚ. Kým muži po Lotyšsku a Estónsku najrýchlejšie ochorejú, ženy sú na tom ešte horšie. Pri porovnávaní očakávanej dĺžky života v plnom zdraví obsadili poslednú priečku. Nakoľko predpokladaná dĺžka života je u mužov 73 rokov a u žien 80, v podstate to znamená, že až štvrtinu, resp. tretinu tohto času trpia zdravotnými problémami.

OECD však uvádza, že sme jednou z krajín, ktoré najviac navštevujú lekárov. Priemerný Slováčik totiž zavíta k lekárovi 12-krát ročne, čo je až dvojnásobok v porovnaní s ostatnými štátmi. Zdravotná starostlivosť na takéhoto pacienta stojí viac ako 1000 € ročne, z čoho tri štvrtiny sa hradia z verejných zdrojov. Výdavky na liečivá u nás dosahujú až dve percentá HDP, vďaka čomu nám patrí štvrtá pozícia v rebríčku v rámci krajín EÚ.

Prečo je tomu tak? Prečo sme chorľavejší ako naši susedia? To je otázka, ktorú pravdepodobne nezodpovieme. Prečo však chodíme tak často k lekárom a platíme viac za lieky? Je to naozaj nevyhnutné a vždy osožné? Sú informácie, na základe ktorých boli vyššie spomínané analýzy, spoľahlivé?

Jednou z možností, ako ušetriť čas a peniaze ľudí aj zdravotníckeho sektora je zvýšenie efektivity poskytovanej zdravotnej starostlivosti. Neraz totiž príde pacient ku svojmu všeobecnému lekárovi, ktorý ho na základe vyšetrenia pošle za rôznymi špecialistami. Pri návšteve prvého mu daný lekár odoberie biologické vzorky a odošle do laboratória, pričom mu prikáže zastaviť sa, keď už budú výsledky z testov. Medzičasom navštívi pacient aj iného špecialistu, u ktorého sa situácia zopakuje. Takýmto spôsobom mu v priebehu niekoľkých dní zopakujú niektoré vyšetrenia. Čiastočným riešením tohto problému je elektronizácia zdravotníctva, konkrétne laboratória online. Tento proces bude šetriť čas pacienta, zníži vyťaženie lekárov a ušetrí financie z verejného



zdravotného poistenia. Vyhli by sme sa aj situáciám, ktoré poistencom viac škodia ako pomáhajú. Totiž veľa pacientov je liečených nadmerným množstvom liekov. Napr. jeden známy prípad panej [4], ktorej bolo v priebehu troch mesiacov predpísaných viac ako 36 druhov liekov od desiatich rôznych špecialistov. Nakoľko o sebe nevedeli, mnohé z týchto liekov mali kontraindikáciu. V snahe pomôcť nakoniec mohli uškodiť.

Aj takýto prípad je motiváciou pre štátnu poisťovňu pre testovanie systém elektronických receptov, službu obdobnú tej, ktorú už využíva zdravotná poisťovňa Dôvera. Elektronická preskripcia dáva včasnú informáciu lekárovi o jeho pacientovi a liekoch, ktoré užíva, no zároveň pomáha poisťovni získať lepšiu kontrolu nad nakladaním s finančne náročnými liekmi. Taktiež vnáša do systému viac transparentnosti a sťažuje falšovanie receptov a vykázanú zdravotnú starostlivosť. A s tým, že sa takéto podvody dejú, majú skúsenosť ako VŠZP, tak aj Dôvera ZP, a.s.

Je viacej možností, ako odstrániť podvody zo zdravotníckeho prostredia a efektívnejšie hospodáriť s jeho prostriedkami. Tou prvou sú systematické zmeny. Pokiaľ by celá zdravotná starostlivosť a jej poskytovanie mala jasne stanovené pravidlá a postupne by sa takmer všetko zelektronizovalo, celý proces by bol transparentnejší a priestor na podvádzanie oveľa menší, ak vôbec nejaký. Táto zmena si však vyžaduje dlhodobý časový horizont, počas ktorého pravdepodobne nastane veľa pokusov a omylov, kým sa nastolia vhodné a najmä rozumné pravidlá.

Druhým spôsobom na priblíženie sa k cieľu, je revízná kontrola. Pokiaľ budeme schopní povedať a predpovedať, kde je priestor na nezákonne obohatenie sa, budeme schopní zakročiť proti už rozbehnutým podvodom a zároveň predchádzať ďalším, novým. Problémom sú hlavne obmedzené kapacity revíznych pracovníkov poisťovne, ktorí dokážu skontrolovať iba malé percento poskytovateľov a preto je dôležité správne zacieliť revíznú činnosť. Momentálne na to slúžia tieto tri zdroje:

1. podnety poistencov, v ktorých z celej zdravotnej starostlivosti vykázané na ich meno, resp. rodné číslo, označia tú, ktorá im nebola poskytnutá,
2. anonymné tipy zo zdravotníckeho prostredia
3. jednoduché analýzy o medziročnom náraste a poklese nákladov u poskytovateľov zdravotnej starostlivosti spojených s intuíciou ako dátových analytikov tak aj

zdravotne vyškoleného personálu

V prvom aj druhom prípade je zdravotná poisťovňa odkázaná na dodatočné informácie od tretej strany. V tom poslednom to tak síce nie je, ale jedná sa naozaj o jednoduché, nie veľmi sofistikované a časovo pomerne náročné spôsoby, ktoré však odhalia len najväčšie neštandardnosti. Preto motiváciou našej diplomovej práce je práve snaha o zefektívnenie a istá algoritmizácia a automatizácia hľadania podvodov.

Nakoľko sa jedná o zdravotníctvo, teda odbor mimo našich štúdií, prvým z našich cieľov je zorientovať sa v jeho terminológii a procesoch, pochopiť fungovanie jednotlivých systémov, odhaliť rizikové oblasti a priestor na podvody. Rovnako dôležité bude osvojiť si prácu s obrovským množstvom dát, nakoľko zdravotná poisťovňa má veľa poistencov, zazmluvnených poskytovateľov zdravotnej starostlivosti a teda aj veľkú databázu o vykázannej starostlivosti. Ďalším cieľom bude nájsť a postupne zefektívniť algoritmy, ktoré by v dátach zdravotnej poisťovne vedeli odhaliť podvody, resp. ktoré by poukázali na neštandardné správanie sa a vytýčili podozrivých poskytovateľov zdravotnej starostlivosti. Tí by následne boli podrobení detailnejšej analýze potvrdzujúcej alebo vyvracajúcej toto podozrenie. Jednou z požiadaviek na takýto algoritmus by mala byť istá univerzálnosť a nenáročnosť, tzn. chceme vytvoriť algoritmus, ktorý by sa spúšťal pravidelne na väčších dátach a bol plne automatizovaný, teda nebola by potrebná žiadna ďalšia úprava dát, či korigovanie algoritmu.

Vzhľadom na tému tejto diplomovej práce venujeme prvú kapitolu zdravotníckej terminológii a dátam zdravotnej poisťovne. Definujeme niektoré pojmy nevyhnutné pre zrozumiteľnosť práce, popíšeme spôsob zberu zdravotníckych dát a niektoré procesy s tým spojené. Nakoniec uvedieme typy podvodov, s ktorými sa zdravotné poisťovne zvyknú stretávať a predpoklady, ako vznikajú. Vzhľadom na nedostatočnú dostupnosť písomných zdrojov, budeme sa opierať najmä o znalosti ľudí pracujúcich v danej oblasti a samotnú databázu zdravotnej poisťovne.

V druhej kapitole najprv uvedieme už existujúci, veľmi jednoduchý algoritmus, spomínaný v [1]. Následne na jeho základoch vybudujeme náš algoritmus. Ten postupne, vzhľadom na jednotlivé požiadavky, upravíme, aby bol čo najefektívnejší a aby sme jeho použitím dostávali želané výsledky.

Tretia kapitola sa bude venovať aplikácii a modifikácii vytvoreného algoritmu. Na

základe už nadobudnutých skúsenosti s podvodmi, spomínanými v [4] a [9], a vzhľadom na naše predpoklady vyberieme vhodné dáta a zanalyzujeme ich. To nám pomôže navrhnúť menšie úpravy a vylepšenia algoritmu, čím sa dopracujeme k želaným výsledkom. Zároveň otestujeme, či boli naše predpoklady správne a či poskytovatelia zdravotnej starostlivosti naozaj podvádzajú opísanými spôsobmi.

Vo štvrtej kapitole uvedieme zopár zaujímavých prípadov, ktoré algoritmus našiel pri spustení na rôzne dáta.

# 1 Zdravotnícka terminológia a dáta

V tejto kapitole spomenieme niekoľko základných pojmov z oblasti zdravotníctva a následne budeme venovať pozornosť dátam zdravotnej poisťovne, odkiaľ pochádzajú, v akom formáte sú a pod. Informácie o poskytovateľoch zdravotnej starostlivosti a zdravotnej starostlivosti ako takej, budeme čerpať najmä z [5],[6],[7] a internej databázy Dôvera zdravotnej poisťovne, a.s. Administratívne a najmä zdravotnícke dáta, spracovávané v tejto práci, sú poskytnuté práve zdravotnou poisťovňou Dôvera ZP, a.s.

## 1.1 Terminológia

Pre potreby tejto práce uvedieme pár termínov, s ktorými sa budeme stretávať pomerne často a preto považujeme za vhodné ich vysvetliť:

- **Zdravotná starostlivosť (ZS)** - všetky zdravotné úkony, tzn. ambulantné i nemocničné vyšetrenia a zákroky, laboratórne vyšetrenia, lieky, zdravotné pomôcky, atď.
- **Poskytovateľ zdravotnej starostlivosti (PZS)** - fyzická či právnická osoba s povolením poskytovať zdravotnú starostlivosť. Formálne sa jedná o systém *hlavnej, vyššej a nižšej zložky*. Hlavná zložka je spravidla jedna, stotožňuje sa s danou fyzickou či právnickou osobou (má pridelené svoje IČO v obchodnom registri, atď). Pod hlavnú zložku patrí zväčša jedna vyššia zložka. Výnimkou sú len niektoré nemocnice (napr. Národný ústav srdcových chorôb), ktoré majú dve vyššie zložky. Nižšie zložky sú už adresné ambulancie, laboratória, lekárne, výdajne zdravotníckych pomôcok, atď., teda konkrétne dvere, na ktoré klopú pacienti. Každá hlavná, vyššia aj nižšia zložka má svoj unikátny kód, pod ktorým vystupuje v databáze poskytovateľov, pričom kód hlavnej zložky je identifikátor poskytovateľa a kód nižšej zložky je kódom poskytovateľa. Na jednotlivých nižších zložkách môže mať úväzok jeden alebo aj viac lekárov, ktorí majú takisto pridelený unikátny kód. Viac o lekároch i poskytovateľoch, tvorbe, systéme a funkcionalite ich kódov nájdeme v [5]. Na adrese [6] sa nachádza zoznam poskytovateľov, spolu s ich kódmi, názvami, IČO a adresami. Zároveň na portáli

Úradu pre dohľad nad zdravotnou starostlivosťou [7] je možné pomocou názvu alebo kódu vyhľadať ostatné informácie o akomkoľvek poskytovateľovi.

- **Vykázaná zdravotná starostlivosť (VZS)** - akákoľvek poskytnutá zdravotná starostlivosť, ktorá je poskytovateľom vykázaná do zdravotnej poisťovne, napríklad za účelom úhrady.
- **odbornosť lekára/nížšej zložky** - rozlišuje o akého lekára/nížšiu zložku sa jedná, napr. všeobecný lekár, hematológ, očný lekár, gastroenterológ, atď.

## 1.2 Dáta zdravotnej poisťovne

Zdravotné poisťovne na Slovensku majú o svojich poistencoch rôzne administratívne dáta, rovnako ako o zazmluvnených poskytovateľoch zdravotnej starostlivosti. Nás však zaujímajú zdravotnícke dáta o nich. Tie sa zhromažďujú na základe vykazovania zdravotnej starostlivosti.

### 1.2.1 Vykazovanie zdravotnej starostlivosti

Každý poskytovateľ zdravotnej starostlivosti má povinnosť zo zákona viesť správny a pravdivý záznam o ním poskytnutej zdravotnej starostlivosti pacientovi. Podľa tejto dokumentácie odosiela minimálne raz za mesiac do zdravotnej poisťovne dávku so zoznamom všetkých výkonov. Na jej základe sa vytvorí faktúra, ktorá je následne hradená zdravotnou poisťovňou.

Jeden riadok dávky obsahuje záznam o jednom výkone. Nachádzajú sa v ňom, resp. mali by sa v ňom nachádzať informácie o:

1. poskytovateľovi zdravotnej starostlivosti, ktorý zdravotnú starostlivosť vykonal
2. lekárovi, ktorý tak urobil
3. samotnom výkone
4. jeho množstve a cene
5. poistencovi, ktorému bola starostlivosť poskytnutá
6. jeho diagnóze, ktorej sa starostlivosť týkala

7. dátume/dátumoch (pokiaľ sa jedná napr. o ambulantné vyšetrenie, dátum je len jeden, ak je to však hospitalizácia v nemocnici, dátumy sú dva a to začiatok a koniec hospitalizácie, rovnako pri predpise liekov a pomôcok sa uvádza dátum predpisu a dátum výdaja)

Pokiaľ poskytovateľ zdravotnej starostlivosti predpísal lieky, pomôcky alebo vyšetrenie u špecialistu, či v laboratóriu, aj táto informácia sa v dátach objaví - v riadku prislúchajúcom k výberu lieku/pomôcky, resp. vykonania vyšetrenia u iného špecialistu/v laboratóriách, bude tento poskytovateľ uvedený ako odosielajúci.

Vzhľadom na to, že Dôvera zdravotná poisťovňa, a.s. má zazmluvnených viac ako 11 000 lekárov, nemocníc a ďalších zdravotných zariadení a poisťuje cca 1,45 miliónov poistencov, každý mesiac prichádzajú dávky s dohromady približne 15 miliónov riadkov. Pri spracovávaní dát za obdobie jedného roka máme dočinenia s cca 114 miliónmi riadkov o zdravotnej starostlivosti.

Všetky prichádzajúce dáta sú spracovávané systémom, ktorý každému jednému riadku na základe vnútorných pravidiel priradí chybu od 1 po 5. Výkon s chybou č. 5 je automaticky zamietnutý, pretože nespĺňa základné pravidlá vykazovania, resp. je v konflikte s inými vykázanými výkonmi. Chyby č. 1 až 2 sú nezávažné a výkony sú systémom akceptované. Ostatné riadky, teda s chybou č. 3 a 4, ostávajú na posúdenie revíznym lekárom a farmaceutom, ktorí zhodnotia správnosť a vhodnosť vykázaných výkonov, často po komunikácii s vykazujúcim poskytovateľom, či ďalším revíznym lekárom, resp. farmaceutom.

Aj keď systém odstraňuje najvýraznejšie nezrovnalosti v dátach a revízni lekári spolu s farmaceutmi chybné vykazovanie, či rôzne kontraindikácie, stále sa jedná spracovanie jedného záznamu s veľmi obmedzeným kontextom k ostatným riadkom vykázaných zdravotnej starostlivosti.

Okrem takéhoto posudzovania dát sa robia aj základné ad hoc analýzy, tzn. nárasty/poklesy nákladov na zdravotnú starostlivosť podľa odborností, skupín až po jednotlivcov pre rôzne účely poisťovne. Nie sme si však vedomí toho, že by boli vytvorené nejaké algoritmy spracovávajúce väčší objem dát za účelom hľadania a odhaľovania podvodov.

### 1.2.2 Typy podvodov a omylov pri vykazovaní

Čo je vlastne podvod v zdravotníckych dátach?

Zdravotná poisťovňa má na základe zmluvy s poskytovateľom zdravotnej starostlivosti povinnosť preplatiť náklady na kompletnú zdravotnú starostlivosť poskytnutú jej poistencom vo výške a množstve vymedzenom v už spomínanej zmluve, až na zopár výnimiek vyplývajúcich zo zákona, viď [8].

Tieto náklady musí ZP znášať a mala by ich financovať z odvodov na povinné zdravotné poistenie každého občana. Mohli by sme to teda zhrnúť pár slovami, a síce, že každá vykázaná zdravotná starostlivosť je zaplatená v súlade s podmienkami zmluvy. Otázkou však ostáva, či každá vykázaná zdravotná starostlivosť bola aj poskytnutá, teda či zo strany poskytovateľa zdravotnej starostlivosti neprichádza aj neoprávnená žiadosť o úhradu zdravotných úkonov.

Pokiaľ nejaký poskytovateľ vykazuje aj neposkytnutú zdravotnú starostlivosť, obohacuje sám seba a ochudobňuje zdravotnícky sektor - kým on si polepší, zdravotná poisťovňa príde o peniaze z vybraného poistného, tým pádom jej ostáva menej prostriedkov dostupných pre ostatných poskytovateľov. Teda koniec koncov prostriedky, ktoré idú neoprávnene na konto poskytovateľa chýbajú práve poistencom.

Prax ukazuje, že pri vykazovaní zdravotnej starostlivosti sa vyskytuje mnoho chýb a omylov, najčastejšie sa stretávame s týmito:

1. vykazovanie väčšieho množstva produktov, ako bolo poskytnuté
2. vykazovanie iných produktov a výkonov, aké boli reálne urobené
3. vykazovanie na poistenca, ktorý poskytovateľa nenavštívil a zdravotná starostlivosť mu poskytnutá nebola

Tieto chyby môžu byť neškodné preklepy či lenivosť zdravotníckeho pracovníka vyhľadať, resp. skontrolovať údaje o poistencovi a poskytovanej zdravotnej starostlivosti, no rovnako môžu indikovať podvod:

1. úmyselné nadmerné vykazovanie
2. vykazovanie drahších liekov a výkonov ako tých, ktoré boli poskytnuté (napr. namiesto lacnej sériovej zdravotníckej pomôcky je vykázaná drahšia individuálna,

resp. komplexnejšie a teda finančne náročnejšie vyšetrenia, atď)

3. vykazovanie naplánovanej zdravotnej starostlivosti, ktorá sa z nejakého dôvodu nakoniec neposkytla
4. fiktívne vykazovanie na poistenca
5. predávanie databáz poistencov

Podvod typu 1 je možné odhaliť jednoduchou analýzou nárastu a poklesu nákladov daného poskytovateľa, je však náročné vzhľadom na ich množstvo, vytypovať daných poskytovateľov. Pokiaľ sa tento podvod deje v rámci istých hraníc, bez tipu od poistenca, či iného človeka s danou skúsenosťou, nie sme schopní detekovať ho.

V prípade 2 a 3 ide o podvody, ktoré sa dajú odhaliť len na základe spolupráce s poistencom. Predchádzať sa im dá (najmä v prípade 2) nastolením jasných pravidiel a kritérií.

Podvodné správanie podľa 4 a 5 je to, na ktoré sa chceme zamerať a nájsť vhodné algoritmy na jeho detekciu.



## 2 Zdieľanie poistencov

Spôsob, ktorým sa chceme na dáta pozerat a bližšie ich analyzovat je zdieľanie pacientov jednotlivými poskytovateľmi zdravotnej starostlivosti. Totižto je úplne bežné, že človek chodí k viacerým lekárom, napr. ku všeobecnému lekárovi, zubárovi a nejakým špecialistom (v závislosti od chorôb, ktoré má). Je teda bežné, že títo dvaja, traja či štyria poskytovatelia zdravotnej starostlivosti budú mať spoločného pacienta. Pokiaľ sa však jedná napríklad o sedem poskytovateľov, ktorí majú spoločné veľké množstvo poistencov, to už je podozrivé. Najmä ak medzi nimi neexistuje nejaké geografické či iné spojivko. Rovnako podozrivé by však bolo, keby dvaja či traja poskytovatelia s rovnakou odbornosťou mali veľa spoločných pacientov - človek zvykne navštevovať len jedného gastroenterológa, otorinolaryngológa či hematológa. V týchto prípadoch vyvstáva otázka, či je vykázaná zdravotná starostlivosť aj naozaj poskytnutá alebo sa jedná o fikciu.

V publikácii [1] sa venujú rovnakej téme. Autori sa zaoberajú americkým programom Medicare, ktorý poskytuje zdravotnú aj finančnú podporu pre viac ako 48 miliónov seniorov a invalidných občanov. Aj keď väčšina zo zainteresovaných poskytovateľov sa správa naozaj čestne, nájdu sa aj podvodníci. Tí stoja systém cca 270 miliárd dolárov ročne, vid [2]. Vzhľadom na množstvo zhromaždených dát sa však dajú spracovať rôzne analýzy, z ktorých jedna je práve zdieľanie poistencov.

Autori [1] uvádzajú prípad chiropraktika z Miami, ktorý si prostredníctvom tretej osoby „najal“ pacientov a vykazoval na nich zdravotnú starostlivosť. Tú samozrejme neposkytol. Ak však títo poistenci boli schopní predať svoje údaje jednej osobe, je predpoklad, že to budú schopní spraviť znova.

Naše predpoklady sa vzhľadom na rozdielnosť systému slovenského zdravotníctva a amerického Medicare programu líšia. Uvažujeme nasledovne: ak sa jeden poskytovateľ správa podvodne a vypisuje fikciu na istú skupinu ľudí, pričom si to doteraz poistovňa a ani jeden z dotýčnych pacientov nevšimol, môže predať túto svoju databázu inému poskytovateľovi a beneficiujú z toho obidvaja.

V článku [1] sa zaoberajú najmä tzv. fantómovými lekárňami. Jedná sa o lekárne vytvárajúce falošné recepty na lieky pre pacientov, ktorých osobné údaje získali rovnakým spôsobom ako už spomínaný chiropraktik, resp. odkúpili tieto informácie od inej strany.

Následne tieto lieky akože vydajú. Teda nemajú žiadne náklady na zaobstarávanie liekov a od zdravotnej poisťovne dostanú úhradu za (ne)poskytnutý liek. Takýchto lekární zvykne byť viac a osobné údaje poistencov medzi sebou pravdepodobne zdieľajú.

Nakoľko zdravotná poisťovňa nemá prostriedky na kontrolu všetkých 2 000 lekární na Slovensku, veľmi by pomohlo, keby sme vedeli vytipovať najpodozrivejšie prípady.

## 2.1 Algoritmus pre dvojice poskytovateľov

Prvým krokom algoritmu uvádzaného v [1] je vytvoriť maticu dummy premenných  $P$ , ktorej riadky prislúchajú konkrétnym poistencom a stĺpce konkrétnym nižším zložkám. Na pozícii  $(i, j)$  je 1 ak  $i$  - ty poistenec niekedy navštívil  $j$  - tu nižšiu zložku, inak 0. Teda:

$$P = (p)_{i,j} = \begin{cases} 1, & \text{ak poistenec bol u PZS} \\ 0, & \text{inak} \end{cases}$$

Označme  $p^i$   $i$ -ty stĺpec matice  $P$ . Potom jednoduchým súčinom

$$\hat{P} = P^T P$$

dostaneme maticu, kde prvok

$$(\hat{p})_{i,j} = \langle p^i, p^j \rangle$$

udáva počet spoločných poistencov nižších zložiek s indexami  $i$  a  $j$ .

Tento algoritmus je veľmi rýchlym a efektívnym nástrojom na zisťovanie počtov poistencov pre dvojice poskytovateľov. Cieľom tejto práce je však vedieť zistiť množstvo zdieľaných pacientov viacerými nižšími zložkami. Naša prvá predstava takéhoto algoritmu logicky vychádza z vyššie spomínaného algoritmu.

## 2.2 Rozšírenie pre ľubovoľne veľké $n$

Majme dané:

1.  $n \in \mathbb{N}$  - pre aké veľké  $n$ -tice nižších zložiek chceme urobiť prienik poistencov
2.  $N \in \mathbb{N}$  - počet všetkých nižších zložiek, ktoré vstupujú do analýzy
3.  $M \in \mathbb{N}$  - počet všetkých poistencov, ktorých berieme do úvahy

4.  $P_{(M \times N)}$  - maticu dummy premenných, ktorá zaznamenáva návštevu pacienta u poskytovateľa (viď vyššie)

Aby sme si zjednodušili vysvetľovanie, definujme  $n$ -rozmerný skalárny súčin vektorov:

**Definícia 2.1.  $n$ -rozmerný skalárny súčin**

Majme množinu indexov  $i_1, i_2, \dots, i_n$  a vektory  $x^{i_1}, x^{i_2}, \dots, x^{i_n} \in R^M, n, M \in \mathbb{N}$ . Potom  $n$ -rozmerným skalárnym súčinom nazveme výraz:

$$\langle x^{i_1}, x^{i_2}, \dots, x^{i_n} \rangle = \sum_{k=1}^m (\prod_{j=1}^n (x_k^{i_j}))$$

Pre takto definovaný  $n$ -rozmerný skalárny súčin je jednoduché nájsť poistencov zdieľaných nižšími zložkami  $i_1, i_2, \dots, i_n$  a síce:  $\langle x^{i_1}, x^{i_2}, \dots, x^{i_n} \rangle$ .

Uvedomme si, že takto definovaný  $n$ -rozmerný skalárny súčin je invariantný vzhľadom na permutácie indexov. Táto vlastnosť je veľmi žiadúca, nakoľko nezáleží, či budeme hľadať spoločných pacientov nižších zložiek  $1, 2, 3, \dots, 10$ , alebo nižších zložiek  $10, 9, 8, \dots, 1$ , v konečnom dôsledku sú to poistenci, ktorí navštívili všetkých týchto poskytovateľov.

Navyše, ak budeme chcieť nájsť zdieľaných pacientov pre všetky možné  $n - tice$  nižších zložiek, zjednoduší nám to prácu, nakoľko namiesto výpočtu prieniku poistencov pre nižšie zložky so všetkými možnými permutáciami indexov nám postačí iba výsledok pre všetky možné kombinácie indexov - teda namiesto  $\frac{N!}{(N-n)!}$  výpočtov ich urobíme len  $\binom{N}{n}$

Pokiaľ nás zaujímajú iba spoloční pacienti pre  $n - tice$  nižších zložiek a nepotrebujeme vedieť nič o  $(n - 1), (n - 2), \dots, 2 - tice$ , postačí, ak prieniky vypočítame iba pre indexy  $i_1 > i_2 > \dots > i_n$ . Ak by sme sa predsa len rozhodli brať do úvahy aj  $(n - 1) - tice$ , budeme hľadať zdieľaných poistencov nižších zložiek s indexmi napr.:  $i_1 \geq i_2 > \dots > i_n$ . Teda všeobecne, pri výsledkoch pre  $n, (n - 1), \dots, (n - k) - tice$ , berieme do úvahy tie indexy, pre ktoré platí:  $i_1 \geq i_2 \geq \dots \geq i_k > i_{k+1} > i_n$ .

Kým pri hľadaní počtu spoločných poistencov pre dvojice nižších zložiek sme vytvárali  $N \times N$  maticu  $\hat{P}$ , v  $n$ -rozmernom prípade skonštruujeme jej ekvivalent, a síce  $n$ -rozmernú  $N \times N \times \dots \times N$  maticu  $\hat{\hat{P}}$ , pričom

$$\hat{\hat{P}}_{i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_{10}} = \langle x^{i_1}, x^{i_2}, x^{i_3}, x^{i_4}, x^{i_5}, x^{i_6}, x^{i_7}, x^{i_8}, x^{i_9}, x^{i_{10}} \rangle$$

Teda rozšírenie algoritmu z [1] pre  $n$ -tice vyzerá nasledovne:

	Algoritmus 1
1.	Zvolíme $n$
2.	Vytvoríme maticu dummy premenných $P$
3.	Zvolíme $k$ , čím určíme, aké najmenšie $(n - k)$ -ice nás zaujímajú
4.	Pre všetky indexy $i_1, i_2, \dots, i_n$ spĺňajúce $i_1 \geq i_2 \geq \dots \geq i_{n-k} > i_{n-k+1} > \dots > i_n$ vypočítame $\langle x^{i_1}, x^{i_2}, \dots, x^{i_n} \rangle$

Tento algoritmus je pekný a jednoduchý, avšak nenapĺňa naše potreby. Totižto, v programe R je maximálny počet prvkov vektora (a teda aj matice) 2 147 483 647. Poisťovňa Dôvera zdravotná poisťovňa, a.s. má zhruba 27 percentný podiel na trhu, čo robí približne 1 450 000 poistencov. Navyše, táto poisťovňa zazmluvňuje poskytovateľov zdravotnej starostlivosti, ktorí majú dohromady 20 000 nižších zložiek. To znamená, že matica  $P$  vytvorená z našich dát by viac ako 10 krát presiahla maximálnu kapacitu. Z daného dôvodu sme museli tento algoritmus upraviť.

### 2.3 Upravený algoritmus

Pri vzniku tohto algoritmu sme prevzali myšlienku predchádzajúceho, pričom sme upustili od vytvárania matice  $P$ . Namiesto nej sme v programe R urobili zoznam nižších zložiek, ktorého každý záznam obsahoval zoznam všetkých poistencov, ktorí ju navštívili, teda vytvorili sme *list of lists*:

```
my_list <- dplyr::data, "NZ", function(x)list(x$POISTENEC))
```

Namiesto  $n$ -rozmerného skalárneho súčtu použijeme funkciu *intersect*, ktorá hľadá prienik dvoch vektorov. Upravený algoritmus vyzerá nasledovne:

Algoritmus 2	
1.	Zvolíme $n$
2.	Vytvoríme zoznam zoznamov $my\_list < -dply(data, "NZ", function(x)list(x$POISTENEC))$
3.	Zvolíme $k$ , čím určíme, aké najmenšie $(n - k)$ -ice nás zaujímajú
4.	Pre všetky indexy $i_1, i_2, \dots, i_n$ spĺňajúce $i_1 \geq i_2 \geq \dots \geq i_{n-k} > i_{n-k+1} > \dots > i_n$ vypočítame $intersect$ z $my\_list[i_1][1], my\_list[i_2][1], \dots, my\_list[i_n][1]$

Ďalšou myšlienkou, ktorá ovplyvnila vývoj posledných dvoch algoritmov, bola nasledovná: Ak počítame prienik všetkých možných  $n$ -tíc nižších zložiek, robíme tak pre  $\binom{N}{n}$  kombinácii, čo môže byť naozaj veľa. Keby sme však urobili najprv dvojice, veľa z nich by malo nulový  $intersect$ , teda by nám podstatne znížili počet  $n$ -tíc, ktoré musíme počítať. Ak by sme vypočítali prienik pre trojice, ubudlo by ich ešte viac. To isté pre štvorice, päťice, ... Každý jeden stupeň je výpočtovo menej náročný ako robiť rovno  $n$ -tice, spolu môžu byť rovnako alebo o niečo zdĺhavejšie. Avšak toto predĺženie výpočtov je vyvážené informačným prínosom takéhoto postupu, a síce informáciami o spoločných poistencov nie len  $n$ -tíc, ale aj  $(n-1)$ -tíc,  $(n-2)$ -tíc, ..., až dvojíc.

Keďže budeme vytvárať od dvojíc postupne po  $n$ -tice, budeme brať vždy unikátne kombinácie indexov, pre ktoré platí:

$$i_1 > i_2 > i_3 > \dots > i_n$$

Tým pádom sa vyhneme tomu, že pri počítaní trojice duplicitne spravíme dvojicu, kde

$$i_1 = i_2 > i_3$$

Dôsledkom tohto je, že danú  $n$ -ticu budeme mať abecedne usporiadanú, napr.  $NZ^1$ ,  $NZ^2$ ,  $NZ^3$ , ...,  $NZ^n$ . Teda ak nejaká  $k$ -tica končí  $NZ^N$ , žiadnu  $(k+1)$ -ticu z nej nespravíme.

Výsledný algoritmus je nasledovný:

Algoritmus 3	
1.	Zvolíme $n$
2.	Vytvoríme zoznam zoznamov: $my\_list < -dply(data, "NZ", function(x)list(x$POISTENEC))$

3.	Vytvoríme si pomocnú premennú <i>pocet</i> , ktorá bude zaznamenávať koľko „jednotíc“, dvojíc, neskôr trojíc, ..., n-tíc s nenulovým <i>intersect</i> sme vytvorili.
4.	Nakoľko „jednotíc“ je $N$ , nastavíme: $pocet[1] < -N$
5.	Vytvoríme <i>data frame</i> <i>ntice_NZ</i> , ktorý bude obsahovať všetky zapísané $k$ -tice pre $k \in \{1, 2, \dots, n\}$
5.	Vytvoríme zoznam <i>intersecty</i> , ktorá bude obsahovať prieniky zapísaných $k$ -tic, pričom na $i$ – <i>tom</i> riadku bude <i>intersect</i> $k$ – <i>tice</i> z <i>ntice_NZ</i> [ $i$ ]
6.	Vyplníme prvých $N$ riadkov „jednoticami“, tzn. nižšími zložkami a síce: $ntice\_NZ[i] < -NZ^i$ , rovnako aj zoznam <i>intersecty</i> podľa bodu 5.
7.	Vytvoríme dvojice: Pre $i$ in $1 : (pocet[1] - 1) :$ Pre $j$ in $(i + 1) : pocet[1] :$ Vypočítame <i>intersect</i> $NZ^i$ a $NZ^j$ , teda $inter < -intersect(intersecty(i), intersecty(j))$ Ak $length(inter) > 0$ dvojicu zapíšeme do $ntice\_NZ[pocet[1 : 2] + 1]$ , jej <i>intersect</i> do $intersecty[pocet[1 : 2] + 1] < -inter$ a zvýšime počet vzniknutých dvojíc: $pocet[2] < -pocet[2] + 1$
8.	Vytvoríme $k$ -tice pre $k \in \{3, 4, \dots, n\}$ : Pre $k$ in $3 : n$ Pre $i$ in $1 : pocet[(k - 1)] :$ Do pomocnej premennej <i>pom</i> uložíme index poslednej nižšej zložky v $(k-1)$ -tici v $i$ - <i>tom</i> riadku zapísaných $(k-1)$ -tíc. Pre $j$ in $(pom + 1) : N :$ Vypočítame <i>intersect</i> $i$ - <i>tej</i> $(k-1)$ -tice a $j$ – <i>tej</i> nižšej zložky a uložíme ho do zoznamu <i>inter</i> : $inter < -intersect(intersecty[pocet[1 : (k - 1)] + i], intersecty[j])$ Ak $length(inter) > 0$ $k$ -ticu zapíšeme do $ntice\_NZ[pocet[1 : k] + 1]$ , jej <i>intersect</i> do: $intersecty[pocet[1 : k] + 1] < -inter$ a zvýšime počet vzniknutých $k$ -tíc: $pocet[k] < -pocet[k] + 1$

Všimnime si, že pri tomto algoritme počítame dosť veľa prienikov, o ktorých už dopredu vieme, že budú prázdne. Predstavme si modelovú situáciu:

Máme 10 rôznych nižších zložiek. Vytvárame najprv dvojice, pričom výstup (*data frame* *ntice\_NZ*) po dvojiciach je nasledovný:

1.	$NZ^1$	$NZ^2$
2.	$NZ^1$	$NZ^3$
3.	$NZ^1$	$NZ^4$
4.	$NZ^1$	$NZ^5$
5.	$NZ^1$	$NZ^6$
6.	$NZ^2$	$NZ^4$
7.	$NZ^2$	$NZ^5$
8.	$NZ^2$	$NZ^{10}$
9.	$NZ^3$	$NZ^4$
	$\vdots$	$\vdots$

Všimnime si, že s  $NZ^1$  vytvorili dvojicu len nižšie zložky  $NZ^2, NZ^3, NZ^4, NZ^5$  a  $NZ^6$ . Preto nemá zmysel ku dvojici  $NZ^1, NZ^2$  pridávať  $NZ^7$  a viac. Teda teoreticky by sme mohli z každej kombinácie v riadku od 1 po 4 spraviť trojicu, pričom pre index tretej zložky  $i_3$  by platilo:

$$1 > i_2 > i_3 > 7$$

Nakoľko v poslednom riadku prislúchajúcom k  $NZ^1$ , teda v piatom riadku tabuľky, máme vo dvojici nižšiu zložku s najväčším indexom, s ktorou má  $NZ^1$  nenulový intersekt, z tejto dvojice už nevznikne žiadna trojica.

Už táto úprava algoritmu by pomerne znížila výpočtovú náročnosť. Avšak so situáciou, ktorá vznikla v modelovom príklade pre  $NZ^2$ , by si nijako neporadila. Preto sme sa rozhodli pre nasledovnú úpravu, a síce:

Vytvoríme dvojice, pričom pre každú nižšiu zložku si budeme pamätať, koľko dvojíc s ňou sme vytvorili (takých, kde je táto NZ ako prvá v poradí), tzn. aký veľký blok z nej vznikol. Teda pre  $NZ^1$  to je 5, pre  $NZ^2$  tri. Následne pri tvorení trojíc budeme pre všetky bloky (pri tvorbe k-tic to bude od 1 po  $\text{pocet}[k-1]$ ), pre každý riadok z daného bloku okrem posledného, brať nižšiu zložku s takým indexom, ktorý sa nachádza v tom istom bloku o riadok, dva, ..., nižšie. Teda v prípade  $NZ^1$ :

Máme blok o veľkosti 5. Z dvojice z prvého riadku:  $NZ^1, NZ^2$ , urobíme trojicu s pomocou nižších zložiek postupne  $NZ^3, NZ^4, NZ^5, NZ^6$ , teda:

1.	$NZ^1$	$NZ^2$	$NZ^3$
2.	$NZ^1$	$NZ^2$	$NZ^4$
3.	$NZ^1$	$NZ^2$	$NZ^5$
4.	$NZ^1$	$NZ^2$	$NZ^6$

Z druhého riadku:  $N^1, NZ^3$ , urobíme nasledovné trojice:

1.	$NZ^1$	$NZ^3$	$NZ^4$
2.	$NZ^1$	$NZ^3$	$NZ^5$
3.	$NZ^1$	$NZ^3$	$NZ^6$

Z tretieho riadku:  $N^1, NZ^4$

1.	$NZ^1$	$NZ^4$	$NZ^5$
2.	$NZ^1$	$NZ^4$	$NZ^6$

A zo štvrtého riadku:  $N^1, NZ^5$

1.	$NZ^1$	$NZ^5$	$NZ^6$
----	--------	--------	--------

Všimnime si, že kým pôvodným algoritmom by sme robili  $8+7+6+5+4 = 30$  trojíc, takto vytvoríme iba 10 trojíc. Teda sme automaticky vylúčili 20 možných trojčlenných kombinácií nižších zložiek, ktorých prienik je však prázdny. Pre vytvorených 10 trojíc musíme následne zistiť počet spoločných poistencov.

Výsledný algoritmus, ktorý pri analýzach použijeme je nasledovný:

	Algoritmus 4
1.	Zvolíme $n$
2.	Vytvoríme zoznam zoznamov: $my\_list < -dply(data, "NZ", function(x)list(x$POISTENEC))$
3.	Vytvoríme si pomocnú premennú <i>pocet</i> , ktorá bude zaznamenávať koľko „jednotíc“, dvojíc, neskôr trojíc, ..., n-tíc s nenulovým intersectom sme vytvorili.



4.	Nakoľko „jednotíc“ je $N$ , nastavíme: $pocet[1] < -N$
5.	Vytvoríme <i>data frame</i> $ntice\_NZ$ , ktorá bude obsahovať všetky zapísané $k$ -tice pre $k \in \{1, 2, \dots, n\}$
5.	Vytvoríme zoznam <i>intersecty</i> , ktorá bude obsahovať prieniky zapísaných $k$ -tic, pričom na $i$ – tom riadku bude <i>intersect</i> $k$ – tice z $ntice\_NZ[i]$
6.	Vyplníme prvých $N$ riadkov „jednoticami“, tzn. nižšími zložkami a síce: $ntice\_NZ[i] < -NZ^i$ , rovnako aj zoznam <i>intersecty</i> podľa bodu 5.
7.	Vytvoríme pomocnú premennú <i>bloky</i> , ktorá bude zaznamenávať, koľko $k$ -tic vznikne z tej ktorej $(k-1)$ -tice
8.	<p>Vytvoríme dvojice:</p> <p>Pre <math>i</math> in <math>1 : (pocet[1] - 1) :</math></p> <p>    Pre <math>j</math> in <math>(i + 1) : pocet[1] :</math></p> <p>        Vypočítame <i>intersect</i> <math>NZ^i</math> a <math>NZ^j</math>, teda</p> <p>        <math>inter &lt; -intersect(intersecty(i), intersecty(j))</math></p> <p>        Ak <math>length(inter) &gt; 0</math></p> <p>            dvojicu zapíšeme do <math>ntice\_NZ[pocet[1 : 2] + 1]</math> ,</p> <p>            jej <i>intersect</i> do <math>intersecty[pocet[1 : 2] + 1] &lt; -inter</math> a zvýšime počet vzniknutých dvojíc: <math>pocet[2] &lt; -pocet[2] + 1</math>, zároveň zvýšime aj <math>bloky[i]</math></p>
9.	<p>Vytvoríme <math>k</math>-tice pre <math>k \in \{3, 4, \dots, n\}</math>:</p> <p>Pre <math>k</math> in <math>3 : n</math></p> <p>    Pre <math>bl</math> in <math>1 : length(bloky)</math></p> <p>        Pre <math>i</math> in <math>1 : (bloky[bl] - 1) :</math></p> <p>            Pre <math>j</math> in <math>(i + 1) : bloky[bl] :</math></p> <p>                Vypočítame <i>intersect</i> <math>i</math>-tej <math>(k-1)</math>-tice z <math>bl</math>-tého bloku a nižšej zložky, ktorá je posledná v kombinácii z <math>j</math>-tého riadku toho istého bloku a uložíme do premennej <i>inter</i></p> <p>                Ak <math>length(inter) &gt; 0</math></p> <p>                    <math>(k)</math>-ticu zapíšeme do <math>ntice\_NZ</math> na nový riadok, jej <i>intersect</i> do nového riadku <i>intersecty</i> a zvýšime počet vzniknutých <math>k</math>-tic: <math>pocet[k] &lt; -pocet[k] + 1</math>, rovnako zvýšime aj <math>bloky[i]</math></p>

Nielenže nám tento algoritmus vyhovuje po teoretickej stránke, aj v praxi dáva rýchlejšie výsledky.

Jednou nevýhodou tohto algoritmu je, že výsledku sú na prvý pohľad duplicitné riadky, a síce: pokiaľ algoritmus zapíše do výsledku nejakú  $n$ -tícu, potom v ňom nájdeme aj  $n$  ( $n-1$ )-tíc, ktoré vzniknú odobratím jednej nižšej zložky z danej  $n$ -tice. Za účelom akéhosi „čistenia výsledkov“ sme navrhli ďalší algoritmus

## 2.4 Čistenie výsledkov

Aby sme sa zbavili už spomínaných „duplicit“, použijeme tento algoritmus:

Pre  $i$  in  $2 : n$   
 vezmeme všetky riadky s  $i$ -ticami, uložíme do premennej *cistenie*  
 pre  $j$  in  $i$   
 vezmeme všetky stĺpce okrem  $i$ -teho z premennej *cistenie* a  
 hľadáme zhodu nejakého riadku z *cistenie* s riadkom s  $(i-1)$ -ticami.  
 Ak sa zhoda nájde,  
 príslušnému riadku s  $(i-1)$ -ticou dáme príznak, že ju do  
 výsledku neberieme.

Takéto čistenie je úplne postačujúce, nakoľko každá  $n$ -tica vyčistí  $n$  ( $n-1$ )-tíc, každá ( $n-1$ )-tica zase  $(n-1)$  ( $n-2$ )-íc, z nich každá  $(n-2)$  ( $n-3$ )-íc, ..., každá trojica 3 dvojice až nakoniec každá z týchto dvojíc dve jednotice.

Čistenie výsledkov však nemusí byť až taký benefit, ako sa na prvý pohľad zdá. Totižto predstavme si, že máme  $n = 10$  a desatica je práve jedna, s tromi spoločnými pacientmi. Pritom jedna z deväťíc, ktorá prislúcha k tejto desatici, by mala až 100 zdieľaných poistencov. Ak však výsledok vyčistíme, táto informácia nám unikne. Pritom, čo je podozrivejšie? Keď desiati poskytovatelia zdravotnej starostlivosti majú troch spoločných pacientov, alebo keď ide o 9 nižších zložiek so 100 zdieľanými poistencami?

Uvedomme si, že tento výsledok by ostal zachovaný, ak by sme spustili najprv algoritmus pre  $n = 9$  spolu s čistením a následne ten istý algoritmus pre  $n = 10$  aj s čistením. Výhodou nášho algoritmu je, že sa dá spustiť pre akékoľvek  $n$ . Ak sú výsledky nedostatočné a chceli by sme v algoritme pokračovať, je jednoduché sa v ňom napojiť a teda nie je nutné spúšťať ho nanovo.

Jednou z možností, ako zachovať takéto extrémne zmeny je mierna úprava čistenia, a síce: pokiaľ sa nájde zhoda medzi riadkom z  $(k-1)$ -tíc a riadkom z  $k$ -tíc orezaných o jeden stĺpec, otestuje sa veľkosť poklesu počtu zdieľaných poistencov. Ak by bol takýto relatívny pokles nad určitou hodnotou, riadku z  $(k-1)$ -tíc by sme nedávali príznak, že ju do výsledku neberieme.

Výhodnosť takéhoto čistenia výsledkov je veľmi diskutabilná a záleží, aký výsledok z danej analýzy očakávame. Pokiaľ táto analýza nedáva veľmi veľa výsledkov, je jednoduchšie a hlavne rýchlejšie, vyhnúť sa čisteniu. Ak však očakávame veľké množstvo výsledkov a sme ochotní nejaké zanedbať, môžeme spustiť aj čistenie, čo nám v konečnom dôsledku ušetrí čas pri vytipovaní si podozrivých poskytovateľov.

## 3 Aplikácia algoritmu

Každá analýza by mala začať dôkladným rozmyslením motivácie, základných predpokladov, výberom dát, použitých prostriedkov a požadovaných výsledkov.

### 3.1 Výber dát a spracovanie

Pred výberom dát treba porozmýšľať v akej forme a rozsahu ich potrebujeme. Je zbytočne vyberať z databázy príliš veľa informácií, nakoľko pri danom množstve riadkov sa jedná o časovo náročnú operáciu. Navyše, aj pri spracovávaní v konkrétnom programe môžu nastať ťažkosti vzhľadom na veľkosť súboru. Na druhej strane, ak uprostred analýzy zistíme, že potrebujeme ďalšiu, dodatočnú informáciu, musíme dáta vybrať a spracovávať nanovo.

Keďže ideme analyzovať návštevy pacientov u poskytovateľov, určite potrebujeme mať v každom riadku dát informáciu o ktorého poskytovateľa (PZS) a poistenca (POI) sa jedná. Nakoľko nám nezáleží, či šlo o 5 alebo 30 návštev, postačí nám unikátna kombinácia *PZS – POI*. Čo sa týka ostatných informácií, treba uvažovať o:

- časovom období, za ktoré chceme robiť analýzu
- údajoch, ktoré o PZS potrebujeme
- údajoch, ktoré potrebujeme o POI
- a údajoch o zdravotnej starostlivosti

#### 3.1.1 Časové obdobie

Ohľadne časového obdobia: zdravotné poisťovne majú dostupné dáta spred roku 2000. Vzhľadom k tomu, že je rok 2017 a v rámci občiansko-právnych sporov je možné uplatniť škodu za najviac 4 roky spätne, môžeme dáta spred roku 2012 považovať za irelevantné. Teda máme k dispozícii dáta za obdobie 2012-2016, tzn. 5 rokov. Dôležité je pouvažovať, aké dlhé obdobie chceme analyzovať. Pokiaľ vezmeme celých 5 rokov, vo výsledkoch sa nám objavia zmeny lekárov spôsobené

- presťahovaním sa

- dovŕšením 18. roku života (napr. zmena všeobecného lekára pre deti a dorast za všeobecného lekára pre dospelých, rovnako pediatrická ortopédia za ortopédiu, atď.)
- zánikom a vznikom nových poskytovateľov zdravotnej starostlivosti

Čím viac skrátime obdobie, tým viac takýchto prípadov eliminujeme. Na druhej strane, nemôžeme vziať príliš krátke obdobie, aby sme dostali dostatočne reprezentatívnu vzorku.

Aby výsledky analýz boli v praxi použiteľné a neboli premlčané, pri vyberaní dát uprednostníme roky 2016 a 2015 pred 2012 a 2013. Dĺžku časového obdobia vyberieme aj podľa množstva riadkov, ktoré tak dostaneme. Teda ak potrebujeme všeobecné dáta, uprednostníme jeden rok, ak sú však dáta špecifické a teda množstvo limitované, môžeme vziať napr. aj obdobie 2015-2016.

### 3.1.2 Údaje o PZS

O poskytovateľoch zdravotnej starostlivosti má poisťovňa množstvo informácií, okrem ňou vykázananej starostlivosti aj administratívne dáta. Avšak, väčšinu z nich nebudeme pri použití algoritmu potrebovať. Ak nejaký poskytovateľ vyjde z analýzy ako podozrivý, následne si vieme pozrieť všetky tieto informácie. Algoritmus v podobe, v akej je, vyžaduje iba nejaký unikátny identifikátor pre každú nižšiu zložku, napr. kód nižšej zložky.

Ďalším údajom, ktorý by mohol byť pri výpočte relevantný, je poloha nižšej zložky. Pokiaľ budeme mať GPS súradnice, na ich základe vieme počítat napríklad vzdušnú vzdialenosť medzi jednotlivými poskytovateľmi, resp. medzi nimi a poistencami.

### 3.1.3 Údaje o POI

Rovnako ako pri poskytovateľoch zdravotnej starostlivosti, aj pri poistencoch nám stačí nejaký identifikátor. Takým je napríklad rodné číslo. Avšak, jedná sa o citlivý osobný údaj a navyše nie je unikátny - dvaja ľudia môžu mať rovnaké rodné číslo. Preto namiesto rodného čísla poistenca budeme používať identifikátor poistenca, čo je unikátne číslo, ktoré má každý poistenec pridelený v poisťovni. Toto číslo je navyše bezvýznamové, teda nedá sa z neho zistiť vek, či pohlavie.

Ak v niektorej analýze budeme potrebovať zisťovať vzdušnú vzdialenosť medzi poistencami a nižšími zložkami, poslúžia nám na to práve GPS súradnice.

#### 3.1.4 Údaje o VZS

Ako sme spomínali v podkapitole 1.2.1, o vykázananej zdravotnej starostlivosti máme v každom riadku veľa informácií. Môžeme podľa nich filtrovať, či bol poskytnutý liek, alebo urobené vyšetrenie, atď. Vzhľadom na to, že sa jedná o obsiahlu problematiku, v tejto práci sa jej venovať nebudeme. Každopádne do budúca pri ďalšom vývoji algoritmu je možné zamerať sa aj na tieto aspekty vykázananej zdravotnej starostlivosti.

### 3.2 Analýza lekární

Motiváciou k tejto analýze nám bola doterajšia skúsenosť s podvádzaním v lekárňach. Na stránke zdravotnej poisťovne Dôvera je uvedený prípad lekárne z okresu Martin, ktorá začala oveľa častejšie fakturovať neštandardné položky, a síce nákladný liek na imunitné ochorenie. Pri kontrole priamo na mieste sa zdokumentovala škoda v hodnote takmer 30000€. Podrobnejšie info možno nájsť na [9]

Zdravotná poisťovňa Dôvera však nie je jediná, ktorá ma takúto negatívnu skúsenosť. V mene Všeobecnej zdravotnej poisťovne sa pre médiá do článku [4] o tom zmienila riaditeľka odboru zdravotného a revízneho: *„Pri revízno-lekárskejších kontrolách sme napríklad v lekární zachytili fiktívne predpísané lieky za jeden mesiac v hodnote 36 tisíc eur, avšak lekár uvedený na recepte ich nenapísal a týchto pacientov nikdy nevidel ani nevyšetril.“*

#### 3.2.1 Na celej VZS

Za rok 2016 tvorí celková vykázaná zdravotná starostlivosť lekárňami a výdajňami zdravotníckych pomôcok zhruba 23,5%. Nejedná sa až o také veľké množstvo dát, preto sme chceli urobiť analýzu za obdobie 2015 až 2016.

Avšak spustenie Algoritmu 4 [str. 25] s nastavením  $n = 10$  nemalo praktické využitie. Nielenže samotný výpočet trval veľmi dlho, množstvo výsledkov bolo neuveriteľne veľké, z toho mnoho bolo bezvýznamných. Napríklad trojice lekární, ktoré zdieľali jedného poistenca. Preto sme sa rozhodli, že budeme brať do úvahy iba intersect väčší ako 5.

Takáto úprava však zanedbá všetky  $n$ -tice s intersectom menším ako 6, tie však môžu byť pre veľké  $n$  zaujímavé. Veľkosť intersectu, ktorý berieme do úvahy však vieme meniť v závislosti od požiadaviek analýzy.

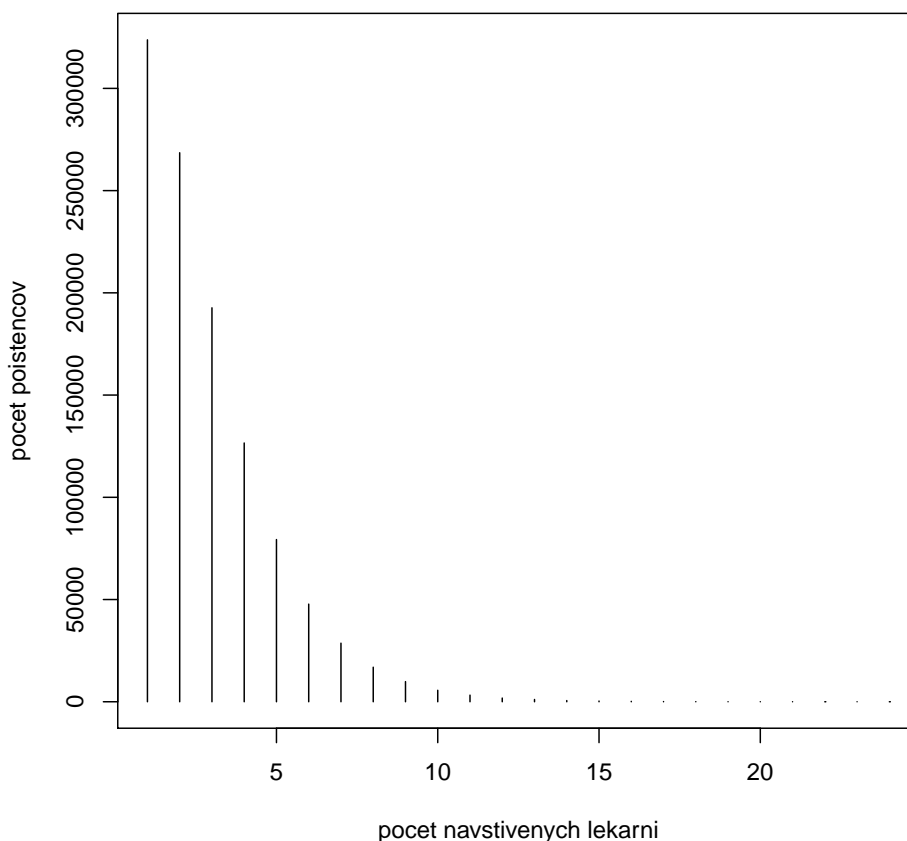
Algoritmus 4 sme upravili v bodoch 8 a 9 nasledovne:

8.	<p>Vytvoríme dvojice:</p> <p>Pre <math>i</math> in <math>1 : (pocet[1] - 1) :</math></p> <p style="padding-left: 2em;">Pre <math>j</math> in <math>(i + 1) : pocet[1] :</math></p> <p style="padding-left: 4em;">Vypočítame intersect <math>NZ^i</math> a <math>NZ^j</math>, teda</p> <p style="padding-left: 4em;"><math>inter &lt; -intersect(intersecty(i), intersecty(j))</math></p> <p style="padding-left: 4em;">Ak <math>length(inter) &gt; 5</math></p> <p style="padding-left: 6em;">dvojicu zapíšeme do <math>ntice\_NZ[pocet[1 : 2] + 1]</math> ,</p> <p style="padding-left: 6em;">jej intersect do <math>intersecty[pocet[1 : 2] + 1] &lt; -inter</math></p> <p style="padding-left: 6em;">a zvýšime počet vzniknutých dvojíc: <math>pocet[2] &lt; -pocet[2] + 1</math>,</p> <p style="padding-left: 6em;">zároveň zvýšime aj <math>bloky[i]</math></p>
9.	<p>Vytvoríme <math>k</math>-tice pre <math>k \in \{3, 4, \dots, n\}</math>:</p> <p>Pre <math>k</math> in <math>3 : n</math></p> <p style="padding-left: 2em;">Pre <math>bl</math> in <math>1 : length(bloky)</math></p> <p style="padding-left: 4em;">Pre <math>i</math> in <math>1 : (bloky[bl] - 1) :</math></p> <p style="padding-left: 6em;">Pre <math>j</math> in <math>(i + 1) : bloky[bl] :</math></p> <p style="padding-left: 8em;">Vypočítame intersect <math>i</math>-tej <math>(k-1)</math>-tice z <math>bl</math>-tého bloku a nižšej zložky, ktorá je posledná v kombinácii z <math>j</math>-tého riadku toho istého bloku a uložíme do premennej <math>inter</math></p> <p style="padding-left: 8em;">Ak <math>length(inter) &gt; 5</math></p> <p style="padding-left: 10em;"><math>(k)</math>-ticu zapíšeme do <math>ntice\_NZ</math> na nový riadok,</p> <p style="padding-left: 10em;">jej intersect do nového riadku <math>intersecty</math> a zvýšime počet vzniknutých <math>k</math>-tíc: <math>pocet[k] &lt; -pocet[k] + 1</math>,</p> <p style="padding-left: 10em;">rovnako zvýšime a <math>bloky[i]</math></p>

Napriek tomu algoritmus objavil veľké množstvo výsledkov, ktoré nebolo možné overiť. Najväčším prekvapením bola stúpajúca tendencia medzi počtami dvojíc, trojíc, štvoríc a päťíc. Očakávanie bolo, že s rastúcim  $n$  bude počet nenulových intersectov klesať. To sa nepotvrdilo, percentuálny podiel nenulových intersectov síce klesal, no

počet možných n-tíc stúpal rýchlejšie.

Pozrime sa, koľko rôznych lekární zvyknú ľudia navštíviť, viď obr. 1. Je na ňom histogram návštevnosti lekární:



Obr. 1: Návštevnosť lekární

Všimnime si, že viac ako 120 tisíc ľudí si v roku 2016 vybrali liek, resp. pomôcku, v štyroch lekárňach. V štyroch a viac lekárňach to je dokonca viac ako 300 tisíc ľudí. Je to viac, ako sme predpokladali. Avšak, nie je to nelogické. Miesta, v ktorých si človek vyberá lieky, ak teda nemá jednu svoju obľúbenú lekárňu sú:

1. blízko svojho lekára
2. pri nemocnici
3. cestou do práce



## 4. doma

Teda hneď štyri rôzne miesta. Keď si uvedomíme, že viac ako 0,6 milióna ľudí v roku 2016 navštívilo najmenej dvoch špecializovaných doktorov, plus každý má svojho všeobecného lekára, lekární „blízko svojho lekára“ môže byť aj viac. Rovnako liek, ktorý poistenec potrebuje, nemusí byť dostupný v každej lekární. Taktiež sa môže jednať o návštevu pohotovostnej lekárne.

S prihliadnutím na predchádzajúce tvrdenia nie je až také prekvapujúce, že vzniká toľko štvoríc, päťíc, . . . , lekární, ktoré zdieľajú viacerých poistencov. Práve táto previazanosť lekární spôsobuje to, že aj keď vykázaná zdravotná starostlivosť lekárňami a výdajňami zdravotníckych pomôcok tvorí len zhruba štvrtinu celkovej zdravotnej starostlivosti, algoritmus bude počítat dlhšie.

Pri analýze lekární a výdajní máme teda štyri možnosti:

1. budeme podozrivých poskytovateľov vyberať z veľkej tabuľky, ktorú nám dal algoritmus spustený na celej vykázanéj zdravotnej starostlivosti lekárňami a výdajňami,
2. zoberieme kratšie obdobie (jeden rok, napr. 2016)
3. odfiltrujeme niektorých poistencov (napr. podľa vzdialeností)
4. rozdelíme databázu na homogénne skupiny produktov, napr. reexportové lieky, inkontinenčné pomôcky, individuálne vyrábané pomôcky, zdravotnícke pomôcky, . . .

### 3.2.2 Rok 2016

Vzhľadom na množstvo výsledkov pri analyzovaní zdravotnej starostlivosti vykázanéj lekárňami a výdajňami zdravotníckych pomôcok za roky 2015 a 2016, rozhodli sme sa spustiť algoritmus len na riadky prislúchajúce k roku 2016. Zároveň každý intersect menší ako 5 považujeme za nulový. Prehľad výsledkov je v tabuľke 1, viď príloha.

Všimnime si, že aj pri analyzovaní vykázanéj zdravotnej starostlivosti lekárňami a výdajňami za obdobie jedného roka dostávame veľké množstvo výsledkov. Síce už nevznikli žiadne desatice, zato dvojíc až sedmíc je mnoho. Pre bližšiu analýzu budeme

vyberať top poskytovateľov z top n-tíc, teda poskytovateľov z pomyselného pravého dolného trojuholníka tabuľky prehľadu. Pred podrobnejšou analýzou ešte zvážime geografické a iné spojenia medzi jednotlivými lekárňami a výdajňami vo vybraných n-ticiach.

Vzhľadom na veľké množstvo n-tíc, ktoré algoritmus vytvoril aj pri analyzovaní zdravotnej starostlivosti vykázanéj lekárňami a výdajňami zdravotníckych pomôcok za jeden rok, skúsime túto zdravotnú starostlivosť ešte rozdeliť.

### 3.2.3 Vzďialenosť ako filter

V nasledujúcich dvoch častiach chceme vykázanú zdravotnú starostlivosť lekárňami a výdajňami zdravotníckych pomôcok rozdeliť na základe vzdialenosti, ktorú musel poistenec prejsť, aby si liek či pomôcku vybral. Tzn., budeme sa pozeráť na vzdialenosť medzi jeho adresou trvalého bydliska a adresou lekárne/výdajne. Z toho dôvodu vyberieme údaje vo forme:

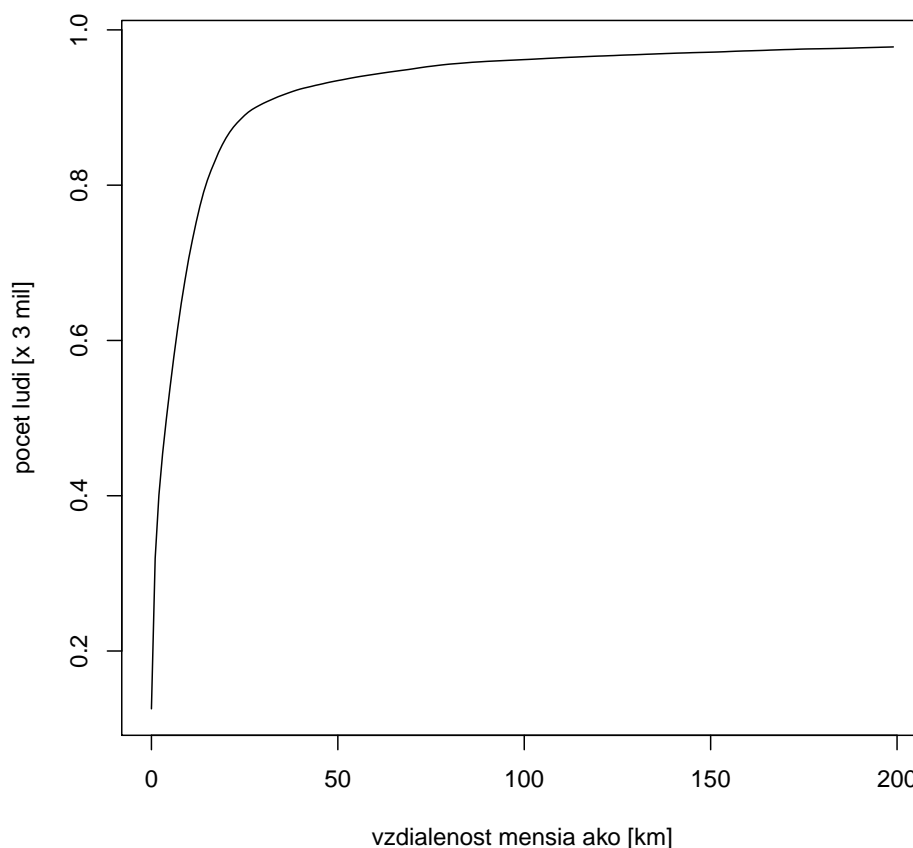
$$KOD\_NZ, GPS\_PZS\_LAT, GPS\_PZS\_LONG, \\ ID\_POISTENEC, GPS\_POI\_LAT, GPS\_POI\_LONG$$

Adresa, resp. GPS súradnice trvalého bydliska je údaj, ktorý vieme vytiahnuť z databázy o poistencoch. Avšak na prvý pohľad to nevyzerá ako najšťastnejšia voľba, keď vezmeme do úvahy napr. ľudí, ktorí cestujú za prácou resp. školou a majú prechodný pobyt, resp. dokonca inú korešpondenčnú adresu. Reálny pohyb poistenca totiž najlepšie vystihuje korešpondenčná adresa. Z databázy zdravotnej poisťovne Dôvera vyplýva, že iba 4% ľudí má korešpondenčnú adresu v inej obci ako trvalý pobyt. Vzhľadom na pomer množstva práce, ktorú by úprava údajov stála, a jej prínosu do analýzy, tento 4%-ný rozdiel zanedbáme.

V tejto časti budeme teda rozlišovať medzi „blízkymi“ a „vzdialenými“ lekárňami. Pozrime sa teda na distribučnú funkciu rozdelenia vzdialenosti medzi adresou trvalého pobytu poistenca a adresou lekárne, viď obr. 2

#### Blízke lekárne

V tejto časti sa budeme zaoberať analýzou „blízkych“ lekárni. Vezmeme tie riadky vykázanéj zdravotnej starostlivosti, kde poistenec prešiel menej ako 35 kilometrov. Toto



Obr. 2: Vzďialenosť lekární a domovov

číslo sme zvolili na základe empirickej distribučnej funkcie rozdelenia vzdialeností medzi adresou trvalého pobytu poistenca a adresou lekárne, viď obr. 2, použitím laktového pravidla.

Motiváciou k takejto analýze je nedávny rozmach predpisovania liekov s psychotropnými účinnými látkami. Jej cieľom je odhaliť skupiny ľudí, ktorí si svojvoľne predpisujú lieky bez vedomia lekára. Predstavme si modelovú situáciu:

Istí ľudia užívajú lieky s psychotropnými látkami nie pre medicínske účely. Nakoľko sa jedná o lieky na predpis, bez receptu si ich nezakúpia, preto použijú falošnú pečiatku lekára a vytvoria falošný recept. Ak by tak robil jeden človek, bol by podozrivý. Zároveň ak by lieky vyberal stále v jednej lekární, vzbudil by podozrenie. Vzhľadom na typ podvodu predpokladáme, že lieky sú vyberané v lekárnach v blízkom okolí. Schéma

podvodu je nasledovná: viacerí poistenci vyberajú lieky vo viacerých lekárňach v jednej oblasti.

Vzhľadom na predchádzajúci rozbor spustíme Algoritmus 4 na dáta zdravotnej starostlivosti vykázanéj lekárňami a výdajňami, pričom z pôvodných dát odstránime riadky, kde vzdialenosť medzi trvalým pobytom poistenca a adresou poskytovateľa prekročila 35 km. Zároveň budeme uvažovať iba intersecty väčšie ako 5, a obdobie rok 2016. Prehľad o výsledných  $n$ -ticiach, intersectoch a ich počtoch je v tabuľke 2 v prílohe. Všimnime si, že deväťice sa nezmenili oproti deväťiciam z predchádzajúcej analýzy (tab. 1), počty  $n$ -tíc pre  $n \in \{2, 3, \dots, 8\}$  však klesli.

### Vzdialené lekárne

Cieľom tejto analýzy bude pokúsiť sa odhaliť organizovaný podvod s fiktívnym vykazovaním liekov a pomôcok na poistencov. V podstate sa pokúsime nájsť lekárne, v ktorých sa môže diať nasledovné: Predstavme si, že lekár v snahe privyrobiť si, dohodne sa s farmaceutom v istej lekárni a začne vypisovať lieky na poistencov. Vytipuje si takých, ktorí nesledujú svoju vykázanú zdravotnú starostlivosť a teda si to nevšimnú (zväčša sa jedná o ťažko chorých ľudí a seniorov). Takto vykázaný liek je samozrejme zdravotnou poisťovňou preplatený. Teda lekár spolu s lekárňou sa protizákonne obohatia.

Takéto fiktívne vykazovanie sa väčšinou nedeje v jednej lekárni, nakoľko daná lekárneň by zaznamenala väčší nárast a nevyhla by sa tak jednoduchej kontrole poisťovňou. Teda tá istá skupina poistencov zrazu začne vyberať lieky aj v inej lekárni. Pokiaľ by sa toto dialo v lekárňach, ktoré sú blízko pacientovi, nie je ľahké to odhaliť, nakoľko je pravdepodobné, že tam bol. Ak by však tieto lekárne boli dostatočne vzdialené, pravdepodobnosť jeho návštevy klesá, tzn. situácia je podozrivejšia.

Z tohto dôvodu sme sa rozhodli spustiť náš algoritmus na dáta zdravotnej starostlivosti vykázanéj lekárňami a výdajňami, kde vzdialenosť medzi poistencom (jeho trvalým bydliskom) a poskytovateľom je väčšia ako 15 km. Opäť budeme brať do úvahy iba lekárne a výdajne s najmenej 6 spoločnými zákazníkmi za rok 2016. Prehľad výsledku je uvedený v prílohe v tabuľke 3. Všimnime si, že nevznikli žiadne  $n$ -tice pre  $n \geq 8$  a množstvo výsledkov je podstatne menšie oproti predchádzajúcej analýze.

### 3.2.4 Zdravotnícke pomôcky

Ako sme už spomínali vyššie, pre rýchlejšie a prehľadnejšie výsledky je výhodné vyfiltrovať produkty alebo skupiny produktov, nad ktorým budeme robiť analýzu. Navyše, je málo pravdepodobne, že v jednej podvodnej schéme sa vyskytnú ako lieky na bolesť hlavy, tak aj plienky aj drahé lieky na liečbu AIDS. Preto sme sa rozhodli použiť algoritmus na časť vydávaných produktov a síce na zdravotné pomôcky.

Táto analýza bude menej náročná, nakoľko stačí, ak na začiatku vyberieme len dáta týkajúce sa zdravotníckych pomôcok. Tie tvoria zhruba 7% z celkového počtu riadkov vykázananej zdravotnej starostlivosti lekárňami a výdajňami. Keďže sa jedná o relatívne malé množstvo dát, algoritmus spustíme na roky 2015-2016. Pri tejto analýze budeme brať do úvahy tie lekárne a výdajne, ktoré majú viac ako troch spoločných zákazníkov. Prehľad o výsledných  $n$ -ticiach pre  $n = 10$  je v tabuľke 4, viď príloha.

Všimnime si, že v tabuľke sú najviac šestice, to preto, že každá sedmica lekární a výdajní zdieľala najviac troch poistencov, ktorí si u nich vybrali nejakú zdravotnícku pomôcku. Pre bližšiu analýzu budeme opäť vyberať top kombinácie pre top  $n$ -tice. Predtým, ako ich budeme analyzovať, zvažíme nejaké spojenia medzi nimi, čo sa týka geografickej polohy, atď.

Tento námet pozerá sa nielen na návštevnosť lekárne ale aj na produkty, resp, skupiny produktov, ktoré sa v nej vyberajú, je veľmi zaujímavý a dal by sa v budúcnosti rozšíriť. Totižto, ak si jeden poistenec vyberie v lekární X lieky proti nádche a v lekární Y antidepresíva, je to viac - menej nezaujímavé. Predsa len, lieky na nádchu sú pomerne lacné a takáto dvojica zbytočne zahľucuje ako algoritmus tak aj výsledok. Ak by však vyberal v obidvoch antidepresíva, to už je výsledok, ktorý nás zaujíma viac.

Výsledky tejto jednoduchej analýzy iba nad pomôckami sú oproti tejto druhej analýze ochudobnené. Kým v tej komplexnejšej by sa do zdieľaných pacientov bral každý, ktorý si v daných lekárňach alebo výdajniach vybral rovnaký produkt, v prvej nám ostanú len tí, čo si vybrali pomôcky. Teda názornejšie: v navrhovanej analýze by medzi spoločných pacientov algoritmus zaradil človeka, ktorý si v obidvoch lekárňach vybral liek na hypertenziu, aj človeka, ktorý si v nich vybral ortézu. V našej analýze zoberie iba toho, čo si vybral ortézu. Avšak takáto úprava algoritmu je už zložitejšia, nakoľko by bolo nutné vytvoriť unikátne kombinácie PZS a POI pre každú skupinu produktov,

čo by vyžadovalo veľa vstupných súborov. Tie by bolo dokonca treba nadefinovať predtým, ako by sme vôbec videli výsledok. Vzhľadom na spomínané skutočnosti takúto úpravu algoritmu v tejto práci robiť nebudeme.

### 3.3 Špeciálna ambulantná starostlivosť

V tomto prípade je pre nás motiváciou nedávne zistenie [9], kedy sa na jednu lekárku v odbore vnútorné lekárstvo sťažovalo viacero poistencov s tým, že u pani doktorky nikdy neboli, resp. boli niekoľko rokov spätne a teda sú si istí, že ich v poslednej dobe nevyšetrila, ako to mali uvedené v elektronickej zdravotnej karte. Po bližšom preskúmaní vykázananej zdravotnej starostlivosti sa zistilo, že niektorí z pacientov v jeden deň navštívili až dve interné lekárky zo svojho okolia, z toho jednu vyššie spomenutú. Ďalším zarážajúcim výsledkom analýzy bolo, že až 300 poistencov Dôvery pravidelne navštevovalo obidve lekárky.

Práve preto sme sa rozhodli zanalyzovať návštevnosť špeciálnych ambulantných starostlivostí. Budeme teda hľadať spoločných pacientov pre poskytovateľov zdravotnej starostlivosti typu ŠAS.

Vzhľadom k tomu, že z celkových dát za rok 2016 vo forme

$$KOD\_NZ, ID\_POISTENEC$$

tvorí menej ako 25% z celkového počtu riadkov, túto analýzu urobíme nad rokmi 2015 a 2016.

Pri tejto analýze urobíme zmenu pri výpočtoch, a síce do kombinácií nižších zložiek nebudeme brať tie, ktoré sú vzdialené do troch kilometrov od seba a ktorých odbornosť je rôzna. Dôvod na to je prostý: ak raz poistenec navštevuje diabetológa a ten ho pošle na vyšetrenie zraku, pričom vo vedľajších dverách je práve ambulancia očnému lekárovi, je vysoko pravdepodobné, že kvôli ošetrovaniu orgánu zraku navštíví práve tohto doktora. Keďže diabetológ často posielal poistencov na očné, títo dvaja budú mať veľký počet spoločných pacientov. Z nášho pohľadu sa však nejedná o podvod a teda nechceme, aby sa to zobrazilo vo výsledku.

Keďže ideme počítať vzdialenosti medzi jednotlivými nižšími zložkami, dáta si vy-

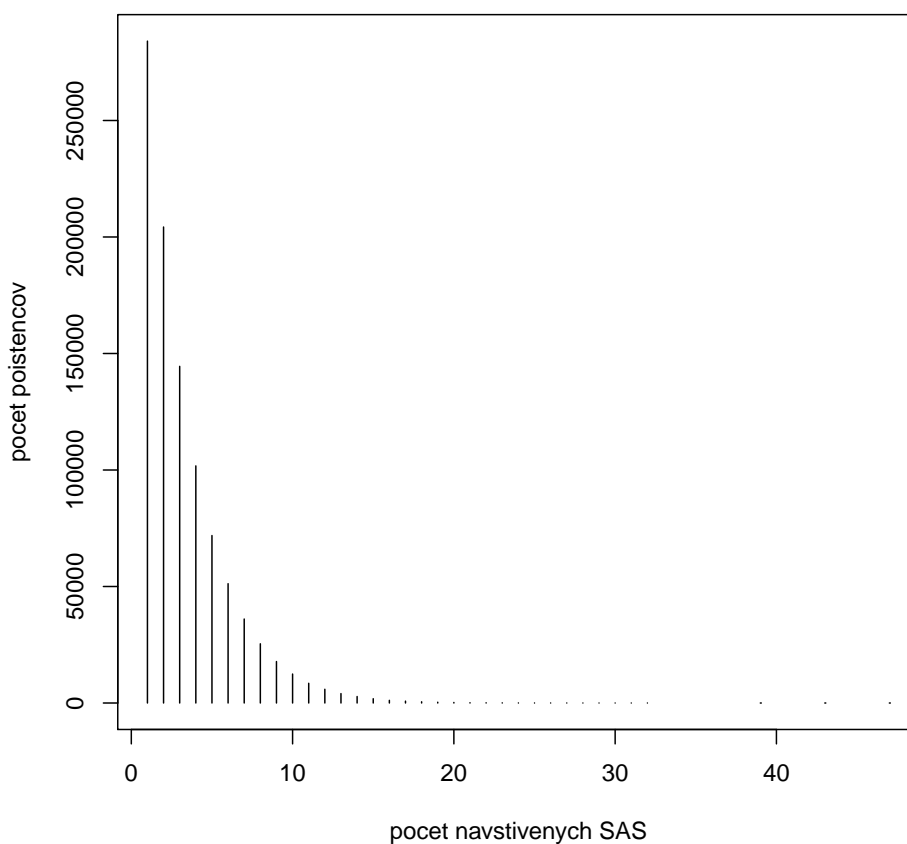
berieme vo forme:

*KOD\_NZ, GPS\_PZS\_LAT, GPS\_PZS\_LONG, ID\_POISTENEC*

Teda okrem kľúčových stĺpcov *KOD\_NZ* a *ID\_POISTENEC* máme aj GPS polohu poskytovateľa.

Mohli by sme do výberu zahrnúť aj odbornosť nižšej zložky, ale tú si môžeme aj jednoducho dopočítať, nakoľko kód nižšej zložky obsahuje kód odbornosti (siedmy až deviaty znak), viď [5]

Ďalšiu zmenu, ktorú v tejto časti urobíme je, že akýkoľvek intersect menší ako alebo rovný 100, budeme považovať za nulový. Totiž, podstatne veľa ľudí navštevuje viacero špecialistov, čo je vidno aj na histograme návštevnosti špecializovaných ambulantných starostlivostí, viď obr.3.



Obr. 3: Návštevnosť ŠAS

Napríklad až takmer 12,5 tisíce ľudí navštívilo práve 10 špecializovaných ambulancií, viac ako 10 ambulancií navštívilo až takmer 39 tisíc. Keby sme teda brali intersect väčší ako 0, vzniklo by nám neskutočné množstvo kombinácií pre  $k \in \{2, 3, \dots, n\}$ . Zároveň, ak aj nájdeme 10 špecialistov, ktorí budú mať troch spoločných pacientov, nie je to podozrivé.

Teda Algoritmus 4 [str.25] upravíme v bode 8 nasledovne:

8.	<p>Vytvoríme dvojice:</p> <p>Pre <math>i</math> in <math>1 : (pocet[1] - 1) :</math></p> <p style="padding-left: 2em;">Pre <math>j</math> in <math>(i + 1) : pocet[1] :</math></p> <p style="padding-left: 4em;">Otestujeme, či <math>NZ^i</math> a <math>NZ^j</math> sú bližšie ako 3 km a zároveň či majú odlišnú odbornosť.</p> <p style="padding-left: 4em;">Ak áno,</p> <p style="padding-left: 6em;">zvýšnú časť tohto bodu preskočíme.</p> <p style="padding-left: 4em;">Ak nie,</p> <p style="padding-left: 4em;">Vypočítame intersect <math>NZ^i</math> a <math>NZ^j</math>, teda</p> <p style="padding-left: 4em;"><math>inter &lt; -intersect(intersecty(i), intersecty(j))</math></p> <p style="padding-left: 4em;">Ak <math>length(inter) &gt; 100</math></p> <p style="padding-left: 6em;">dvojicu zapíšeme do</p> <p style="padding-left: 6em;"><math>ntice\_NZ[pocet[1 : 2] + 1]</math>, jej intersect</p> <p style="padding-left: 6em;">do <math>intersecty[pocet[1 : 2] + 1] &lt; -inter</math> a</p> <p style="padding-left: 6em;">zvýšime počet vzniknutých dvojíc:</p> <p style="padding-left: 6em;"><math>pocet[2] &lt; -pocet[2] + 1</math>, zároveň</p> <p style="padding-left: 6em;">zvýšime aj <math>bloky[i]</math></p>
----	--

Rovnako musíme upraviť aj deviaty bod Algoritmu 4:



```

9. Vytvoríme  $k$ -tice pre  $k \in \{3, 4, \dots, n\}$ :
   Pre  $k$  in  $3 : n$ 
     Pre  $bl$  in  $1 : length(bloky)$ 
       Pre  $i$  in  $1 : (bloky[bl] - 1) :$ 
         Pre  $j$  in  $(i + 1) : bloky[bl] :$ 
           Otestujeme, či posledná zložka  $i$ -tej  $(k-1)$ -tice
           z  $bl$ -tého bloku a nižšia zložka, ktorá je posledná
           v kombinácii z  $j$ -tého riadku toho istého bloku sú
           bližšie ako 3 km a zároveň majú odlišnú odbornosť.
           Ak áno
             zvyšnú časť tohto bodu preskočíme.
           Ak nie,
             Vypočítame intersect  $i$ -tej  $(k-1)$ -tice z  $bl$ -tého
             bloku a nižšej zložky, ktorá je posledná v
             kombinácii z  $j$ -tého riadku toho istého bloku a
             uložíme do premennej inter
             Ak  $length(inter) > 100$ 
                $(k)$ -ticu zapíšeme do ntice_NZ na nový riadok,
               jej intersect do nového riadku intersecty a
               zvýšime počet vzniknutých  $k$ -tíc:
                $pocet[k] < -pocet[k] + 1,$ 
               rovnako zvýšime a  $bloky[i]$ 

```

Takto upravený algoritmus spustíme pre  $n = 10$  na výbere dát popísanom vyššie. Prehľad  $n$ -tíc s počtami zdieľaných poistencov popisuje tabuľka 5, viď príloha.

Môžeme si všimnúť, že tabuľka obsahuje prehľad veľkosti prienikov len pre dvojice, trojice a štvorice. To preto, že všetky možné päťice nižších zložiek zdieľajú menej ako 100 poistencov.

Pri bližšej analýze týchto výsledkov však vidíme, že v mnohých  $n$ -ticiach sa nachádzajú kombinácie nižších zložiek toho istého poskytovateľa zdravotnej starostlivosti

s rovnakou odbornosťou. To znamená, že ak má jeden poskytovateľ dve ambulancie, napr. ortopédia, z ktorých jedna funguje pondelok až streda a druhá od štvrtku do piatku (jav vídaný napr. v nemocniciach), tieto dve nižšie zložky majú veľký prienik a objavujú sa vo výsledku. Dve štvorice, ktoré nám v analýze vznikli sú toho krásnym príkladom.

V jednej sú všetky štyri nižšie zložky patriace jednému poskytovateľovi a majú aj rovnakú odbornosť. Teda takáto kombinácia nás vonkoncom nezaujíma a nemala by vzniknúť (ani žiadna trojica či dvojica z nej). Druhá štvorica obsahuje dvoch poskytovateľov, ktorí obidvaja majú po dve nižšie zložky, tiež s rovnakou odbornosťou. Teda ani jedna z týchto štvoric nie je „ozajstnou štvoricou“. Čo sa týka dvojíc, tam viac ako 900 z celkovo cez 5500 dvojíc tvoria nižšie zložky jedného poskytovateľa s rovnakou odbornosťou. Navyše, majú spomedzi všetkých najvyšší prienik a teda pri pohľade na výsledok by sme ich brali za vhodných kandidátov na preskúmanie. Avšak pre nás sú irelevantné, nakoľko sa nejedná o podvod.

Tieto duplicity môžeme vyriešiť nasledovným spôsobom: do analýzy nebude ako identifikátor poskytovateľa vstupovať kód nižšej zložky, ale kombinácia kódu vyššej zložky a odbornosti - teda nižšie zložky s rovnakou odbornosťou sa spoja do jednej, ale naprieč odbornosťami ostanú samostatne. Takýmto spôsobom budeme mať za obdobie 2015-2016 zhruba o šestinú unikátnych identifikátorov menej, teda o 989 kódov.

Ďalším neželaným javom, ktorý sa vo výsledku objavuje, sú kombinácie obsahujúce transformácie. Občas sa stáva, že nejaký poskytovateľ zdravotnej starostlivosti sa transformuje, napr. z fyzickej osoby sa zmení na právnickú, alebo preberie ambulanciu nový lekár, tým pádom má pridelené nové IČO a teda aj nový kód od hlavnej až po nižšie zložky. V takomto prípade sa v podstate zmení len jeho právna forma a identifikátor PZS, ale tak ako ambulancie tak aj pacienti ostávajú rovnakí. Teda sa objavujú aj vo výsledku.

Nakoľko zdravotná poisťovňa Dôvera si vedie databázu takýchto transformácií, pri hľadaní spoločných poistencov nebudeme uvažovať kombinácie, kde sa nachádza poskytovateľ pred transformáciou aj poskytovateľ po transformácii. Teda ak by mala vzniknúť štvorica:

$$PZS1, PZS2, PZS3_{pred\ transformáciou}, PZS3_{po\ transformácii}$$

, namiesto nej sa vo výsledku objavia iba dve trojice:

$$PZS1, PZS2, PZS3_{pred \text{ transformáciou}}$$

$$PZS1, PZS2, PZS3_{po \text{ transformácii}}$$

Teda sa nám vo výsledku „redukuje stupeň“.

Algoritmus 4 teda modifikujeme v bodoch 8 a 9 nasledovne:

8.	<p>Vytvoríme dvojice:</p> <p>Pre <math>i \text{ in } 1 : (\text{pocet}[1] - 1) :</math></p> <p>Pre <math>j \text{ in } (i + 1) : \text{pocet}[1] :</math></p> <p>Otestujeme, či <math>VZ\_ODB^i</math> a <math>VZ\_ODB^j</math> sú bližšie ako 3 km a zároveň či majú odlišnú odbornosť.</p> <p>Ak áno,</p> <p style="padding-left: 20px;">zvyšnú časť tohto bodu preskočíme.</p> <p>Ak nie,</p> <p>Otestujeme, či <math>VZ\_ODB^i</math> a <math>VZ\_ODB^j</math> sú kódy jedného PZS pred a po transformácii.</p> <p>Ak áno,</p> <p style="padding-left: 20px;">zvyšnú časť tohto bodu preskočíme.</p> <p>Ak nie,</p> <p>Vypočítame <math>\text{intersect } VZ\_ODB^i</math> a <math>VZ\_ODB^j</math>, teda <math>\text{inter} &lt; -\text{intersect}(\text{intersecty}(i), \text{intersecty}(j))</math></p> <p>Ak <math>\text{length}(\text{inter}) &gt; 100</math></p> <p style="padding-left: 20px;">dvojicu zapíšeme do <math>\text{ntice\_NZ}[\text{pocet}[1 : 2] + 1]</math>, jej <math>\text{intersect}</math> do <math>\text{intersecty}[\text{pocet}[1 : 2] + 1] &lt; -\text{inter}</math> a zvýšime počet vzniknutých dvojíc: <math>\text{pocet}[2] &lt; -\text{pocet}[2] + 1</math>, zároveň zvýšime aj <math>\text{bloky}[i]</math></p>
----	---

<p>9. Vytvoríme <math>k</math>-tice pre <math>k \in \{3, 4, \dots, n\}</math>:</p> <p>Pre <math>k</math> in <math>3 : n</math></p> <p style="padding-left: 2em;">Pre <math>bl</math> in <math>1 : length(bloky)</math></p> <p style="padding-left: 4em;">Pre <math>i</math> in <math>1 : (bloky[bl] - 1) :</math></p> <p style="padding-left: 6em;">Pre <math>j</math> in <math>(i + 1) : bloky[bl] :</math></p> <p style="padding-left: 8em;">Otestujeme, či posledná <math>VZ\_ODB</math> <math>i</math>-tej <math>(k-1)</math>-tice z <math>bl</math>-tého bloku a <math>VZ\_ODB</math>, ktorá je posledná v kombinácii z <math>j</math>-tého riadku toho istého bloku sú bližšie ako 3 km a zároveň majú odlišnú odbornosť.</p> <p style="padding-left: 8em;">Ak áno,</p> <p style="padding-left: 10em;">zvyšnú časť tohto bodu preskočíme.</p> <p style="padding-left: 8em;">Ak nie,</p> <p style="padding-left: 10em;">Otestujeme, či posledná <math>VZ\_ODB</math> <math>i</math>-tej <math>(k-1)</math>-tice z <math>bl</math>-tého bloku a <math>VZ\_ODB</math>, ktorá je posledná v kombinácii z <math>j</math>-tého riadku toho istého bloku sú kódy jedného PZS pred a po transformácii.</p> <p style="padding-left: 8em;">Ak áno,</p> <p style="padding-left: 10em;">zvyšnú časť tohto bodu preskočíme.</p> <p style="padding-left: 8em;">Ak nie,</p> <p style="padding-left: 10em;">Vypočítame intersect <math>i</math>-tej <math>(k-1)</math>-tice z <math>bl</math>-tého bloku a nižšej zložky, ktorá je posledná v kombinácii z <math>j</math>-tého riadku toho istého bloku a uložíme do premennej <i>inter</i></p> <p style="padding-left: 10em;">Ak <math>length(inter) &gt; 100</math></p> <p style="padding-left: 12em;"><math>(k)</math>-tícu zapíšeme do <i>ntice_NZ</i> na nový riadok, jej intersect do nového riadku <i>intersecty</i> a zvýšime počet vzniknutých <math>k</math>-tíc: <math>pocet[k] &lt; -pocet[k] + 1</math>, rovnako zvýšime a <math>bloky[i]</math></p>
---

Spustením tohto modifikovaného algoritmu dostávame výsledok, ktorého prehľad je v

tabulke 6, viď príloha.

Všimnime si, že po spustení upraveného algoritmu nám ostala už len jedna štvorica, počet trojíc sa nám zredukoval z 329 na 248 a počet dvojíc nám stúpol z 5307 na 7149. Vznikajú otázky:

1. Prečo máme štvoricu, keď sme predtým mali dve, z ktorých ani jedna nebola „ozajstnou štvoricou“?

Pri analýze, kde sme ako identifikátor poskytovateľa brali kód nižšej zložky, nám táto „štvorica“ poskytovateľov ostala medzi trojicami, kde vždy jeden poskytovateľ (v podobe nižšej zložky) chýbal - spolu totiž nespĺnili podmienku intersectu väčšieho ako 100. Keď sme však všetky nižšie zložky poskytovateľa s rovnakou odbornosťou spojili do jednej, pri jednom z týchto štyroch poskytovateľov sme spojili až 4. Teda sme spojili aj všetkých jeho pacientov. Takto urobený intersect už bol väčší ako 100, konkrétne rovný 115 a teda štvoricu vytvorilo.

2. Prečo nám tak narástol počet dvojíc?

Všimnime si, že z trojice sme zlúčením nižších zložiek s rovnakou odbornosťou dokázali vyrobiť štvoricu, nakoľko pri jednom poskytovateľovi sa zväčšil počet pacientov. Toto sa stalo odhadom pri viac ako 300 poskytovateľoch. Teda keď niektoré nižšie zložky neboli schopné vytvoriť dvojice, zlúčené do jednej už áno. A tu vzniká to množstvo dvojíc. Navyše, uvedomme si, že naše pôvodné nie „ozajstné“ štvorice sa stupňovo redukovali, jedna na „jednoticu“, druhá na dvojicu. A aj tu je dôvod nárastu počtu dvojíc. Keď vezmeme do úvahy, že 2 štvorice a takmer 100 trojíc sa redukovalo na „jednotice“ a dvojice, ich nárast je logický.

## 4 Zaujímavé výsledky

V tejto kapitole uvidíme zopár zaujímavých prípadov, ktoré sme našli vo výsledkoch vytvorených našim algoritmom a ktoré sú vhodné na ďalšiu analýzu.

Nakoľko analýzy zdravotnej starostlivosti vykázanéj lekárňami a výdajňami zdravotníckych pomôcok skončili zväčša väčším množstvom výsledkov, je náročnejšie určiť podozrivých poskytovateľov. Najzaujímavejšie výsledky máme z časti analyzovania tzv. „ďalekých lekární“, kde je napríklad šesticca lekární z okresov Košice, Košice-okolie, Spišská Nová Ves a Prešov zdieľajúca šiestich poistencov.

Rovnako aj päť lekární so sídlami v Košickom, Rožňavskom a Trenčianskom okrese, ktoré majú 21 spoločných pacientov. Táto lekárňa z Trenčína má veľa nenulových prienikov s východoslovenskými lekárňami z okresu Košice, Košice-okolie, Rožňava, Spišská Nová Ves, rovnako tak z okresu Nové Mesto nad Váhom a Bratislava.

Zaujímavá je aj lekárňa z Košíc, ktorá zdieľa pacientov s lekárňami z Michaloviec, Rimavskej Soboty, Brezna, Martina, Žiliny, Čadce, Námestova, Púchova, Nových Zámkov, Zvolena, Krupiny, Levíc, ... a nejedná sa o lekárňu pri nemocnici, či poliklinike.

Pri analyzovaní špeciálnej ambulantnej starostlivosti nám vyšla jedna štvorica so 115 spoločnými pacientmi. Najzaujímavejšie na nej je, že všetci štyria poskytovatelia sú chirurgické ambulancie. Z toho jedna nižšia zložka je v rámci nemocnice, ostatné sú súkromné. Ďalším zaujímavým faktom je, že títo traja súkromní poskytovatelia sú z jedného menšieho mesta (cca 18 000 obyvateľov). Navyše, ak z tejto štvorice odoberieme nižšiu zložku z nemocnice, počet spoločných pacientov sa takmer strojnásobí. Dokonca keď z tejto trojice urobíme tri dvojice (odoberieme jednu nižšiu zložku), tie zdieľajú postupne 810, 1 250 a 1 500 poistencov.

Okrem vyššie spomenutej trojice je vo výsledku veľa zaujímavých dvojíc. Napríklad dve dermatovenerologické ambulancie na juhu Slovenska. Delí ich menej ako 15 kilometrov a zdieľajú cez 600 poistencov.

Na východe Slovenska zase ordinujú blízko seba dvaja otorinolaryngológovia, ktorí zdieľajú 550 pacientov.

## Záver

Vzhľadom na aktuálnosť témy finančného stavu zdravotníctva je detekcia podvodov v dátach zdravotnej poisťovne vhodným priestorom na prieskum a algoritmicizáciu.

Hlavným cieľom tejto diplomovej práce bolo navrhnúť a optimalizovať algoritmus, ktorý by z veľkého množstva poskytovateľov zdravotnej starostlivosti vytýčil podozrivo sa správajúcich. Vychádzajúc z jednoduchého, už existujúceho algoritmu bol v kapitole 2 navrhnutý a postupne vybudovaný nový algoritmus. Následne sa v kapitole 3 modifikoval pre potreby konkrétnych analýz a na rôznych výberoch dát sa demonštrovala jeho široká využiteľnosť. Poskytovatelia zdravotnej starostlivosti, ktorých algoritmus vytipoval, sa podrobnejšie prezreli a niektorí z nich boli označení za vhodných adeptov na hlbšiu analýzu.

Najväčším prínosom tejto práce a vytvorenia tohto algoritmu je schopnosť vytipovať podozrivo sa správajúcich poskytovateľov zdravotnej starostlivosti a tak presnejšie a efektívnejšie zacieliť revíziu kontrolu, čím sa zefektívni využívanie zdrojov zdravotnej poisťovne v tejto oblasti. Široká aplikovateľnosť tohto algoritmu zjednoduší prácu ako dátovým analytikom, tak aj revíznym lekárom a farmaceutom.

Hlavným prínosom práce pre autora je nadobudnutie poznatkov z fungovania zdravotnej poisťovne a rovnako práce s veľkým množstvom dát. Zároveň pomohla vytvoriť si systém pri analyzovaní dát, od prvotného rozboru problematiky, cez dátový výber až po samotnú realizáciu analýzy.

Nakoľko je táto práca jedna z mála svojho druhu, ponúka veľký priestor na prípadné pokračovanie v danej problematike. Načrtla niekoľko smerov, ktorými sa dá uberať pri úpravách samotného algoritmu - či už istý spôsob čistenia výsledkov, alebo možnosť meniť veľkosť signifikantného intersectu s rastúcim  $n$ , respektíve ísť pri analýze lekární a výdajní zdravotníckych pomôcok o úroveň nižšie a sledovať aj druhy vykazovaných produktov.

## Zoznam použitej literatúry

- [1] Gangopadhyay, A., Chen, S., Yesha, Y.: *Detecting Healthcare Fraud through Patient Sharing Scheme*, University of Maryland Baltimore County, dostupné na internete (13.05.2017): [link.springer.com/chapter/10.1007/978-3-642-29166-1\\_39](http://link.springer.com/chapter/10.1007/978-3-642-29166-1_39)
- [2] The Economist: *The \$272 billion swindle*, dostupné na internete (13.05.2017): [www.economist.com/news/united-states/21603078-why-thieves-love-americas-health-care-system-272-billion-swindle](http://www.economist.com/news/united-states/21603078-why-thieves-love-americas-health-care-system-272-billion-swindle)
- [3] Krčmárik, R.: *K lekárom chodíme aj s maličkostami. A až privela si priplácame*, dostupné na internete (13.05.2017): <https://aktualne.atlas.sk/slovensko/zdravotnictvo/k-lekarom-chodime-aj-malickostami-privela-priplacame.html>
- [4] TASR: *Štátna poisťovňa testuje novinku. Systém elektronických receptov na lieky*, dostupné na internete (13.05.2017): <https://aktualne.atlas.sk/slovensko/zdravotnictvo/statna-poisťovna-testuje-novinku-system-elektronickych-receptov-lieky.html>
- [5] ÚDZS: *Metodické usmernenie k pridelovaniu číselných kódov č. 13/2014*, dostupné na internete (13.05.2017): <http://www.udzs-sk.sk/documents/14214/19493/MU+č.+13-2014-Pridelovanie+číselných+kódov+ZP%2C+PZS%2C+zariadenia+soc.+služieb+a+soc.právnej+ochrany+detí+a+soc.+kuratery.pdf/04347e62-44f3-4351-a908-2614c34c3ca7>
- [6] ÚDZS: *Kódy poskytovateľov zdravotnej starostlivosti a zdravotníckych pracovníkov*, dostupné na internete (13.05.2017): <http://www.udzs-sk.sk/kody-poskytovatelov-zdravotnej-starostlivosti-a-lekarov>
- [7] ÚDZS: *Overenie poskytovateľa zdravotnej starostlivosti*, dostupné na internete (13.05.2017): <https://www.portaludzs.sk/overenie-poskytovateľa-zdravotnej-starostlivosti>



- [8] Dôvera ZP, a.s.: *Zdravotná starostlivosť a jej pravidlá*, dostupné na internete (13.05.2017): <https://www.dovera.sk/poistenec/zdravotna-starostlivost>
- [9] Dôvera ZP, a.s.: *Prípady faktúrovania fiktívnej zdravotnej starostlivosti*, dostupné na internete (13.05.2017) : <https://www.dovera.sk/press/informacie-pre-media/a2106/pripady-fakturovania-fiktivnej-zdravotnej-starostlivosti>

## Príloha

Tabuľka 1: Lekárne za rok 2016

dlzka intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice	osmice	devatice
6	8498	35303	38563	24517	10052	2117	184	4
7	6915	25612	26672	16274	6140	1087	77	2
8	5532	19795	19592	11278	3977	638	39	1
9	4491	15624	14894	8410	2656	372	22	0
10	3878	12587	11579	6266	1913	224	9	0
11	3410	10493	9204	4765	1426	165	5	0
12	2921	8863	7632	3913	1014	110	4	0
13	2713	7550	6417	3095	808	64	4	0
14	2364	6406	5245	2491	592	44	1	0
15	2116	5577	4566	2121	481	35	1	0
16	1881	4830	3786	1693	401	32	1	0
17	1725	4293	3324	1410	315	17	0	0
18	1571	3883	2877	1248	241	17	0	0
19	1524	3578	2548	1110	213	12	0	0
20	1389	3083	2276	951	164	12	0	0
21	1253	2860	1938	789	109	7	0	0
22	1235	2539	1776	673	108	3	0	0
23	1200	2350	1640	618	96	6	0	0
24	1072	2136	1536	528	58	2	0	0
25	987	1928	1341	497	72	4	0	0
26	946	1827	1175	445	52	2	0	0
27	870	1654	1116	376	45	2	0	0
28	859	1554	994	341	44	2	0	0
29	783	1467	895	303	39	0	0	0
30	754	1322	871	290	47	0	0	0
31	728	1241	756	280	28	0	0	0

dlzka intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice	osmice	devatice
32	694	1145	728	237	11	1	0	0
33	646	1077	650	218	13	0	0	0
34	588	970	596	218	22	0	0	0
35	593	930	548	167	17	0	0	0
36	541	867	562	153	22	2	0	0
37	518	876	491	148	10	1	0	0
38	543	817	443	136	9	0	0	0
39	482	761	455	96	6	0	0	0
40	472	690	430	106	6	1	0	0
41	455	689	357	91	10	0	0	0
42	458	685	382	96	6	0	0	0
43	419	586	360	88	8	0	0	0
44	448	625	315	93	2	0	0	0
45	375	564	282	78	5	0	0	0
46	377	540	292	75	1	0	0	0
47	410	509	283	69	2	0	0	0
48	347	457	246	55	1	0	0	0
49	352	473	235	67	5	0	0	0
50	340	482	225	51	3	0	0	0
51	339	427	212	43	2	0	0	0
52	328	463	219	47	5	0	0	0
53	317	395	217	44	2	0	0	0
54	295	368	210	27	2	0	0	0
55	283	349	205	39	0	0	0	0
56	280	328	181	23	3	0	0	0
57	244	318	153	36	1	0	0	0
58	287	284	148	37	1	0	0	0
59	250	286	152	26	1	0	0	0
60	258	324	155	27	1	0	0	0
61	255	315	138	32	1	0	0	0

dlzka intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice	osmice	devatice
62	239	285	135	21	1	0	0	0
63	224	263	131	19	3	0	0	0
64	253	264	132	20	0	0	0	0
65	222	271	123	14	0	0	0	0
66	217	210	125	20	1	0	0	0
67	214	259	121	16	1	0	0	0
68	214	211	93	19	1	0	0	0
69	201	213	87	13	0	0	0	0
70	204	219	102	18	1	0	0	0
71	190	209	87	13	0	0	0	0
72	198	195	67	12	0	0	0	0
73	177	208	85	15	0	0	0	0
74	152	194	76	12	1	0	0	0
75	161	163	64	5	0	0	0	0
76	180	183	70	15	0	0	0	0
77	158	182	85	13	0	0	0	0
78	150	167	67	9	0	0	0	0
79	156	179	56	7	0	0	0	0
80	149	151	69	12	1	0	0	0
81	169	142	72	8	0	0	0	0
82	161	129	58	5	0	0	0	0
83	141	147	69	7	0	0	0	0
84	130	126	55	4	1	0	0	0
85	145	151	56	7	0	0	0	0
86	142	120	67	5	1	0	0	0
87	121	148	58	2	0	0	0	0
88	131	96	44	6	0	0	0	0
89	139	112	60	5	0	0	0	0
90	124	134	39	3	0	0	0	0
91	139	102	50	4	0	0	0	0

dlzka intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice	osmice	devatice
92	124	112	40	2	0	0	0	0
93	119	132	45	2	0	0	0	0
94	127	116	42	3	1	0	0	0
95	111	118	53	5	0	0	0	0
96	99	87	54	5	0	0	0	0
97	102	96	33	4	0	0	0	0
98	105	105	40	0	0	0	0	0
99	135	99	46	3	0	0	0	0
100	110	93	29	3	0	0	0	0
101	81	98	30	3	0	0	0	0
102	94	115	31	2	0	0	0	0
103	99	93	26	1	0	0	0	0
104	81	80	28	3	0	0	0	0
105	93	81	26	1	0	0	0	0
106	105	83	28	3	0	0	0	0
107	89	75	26	1	0	0	0	0
108	105	87	34	2	0	0	0	0
109	88	91	37	1	0	0	0	0
110+	9446	5191	1054	49	0	0	0	0

Tabuľka 2: Lekárne 2016 do 35km

dlzka intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice	osmice	devatice
6	4974	27127	30621	19836	8196	1716	138	4
7	4212	20082	21561	13269	5154	888	61	2
8	3448	15901	15876	9327	3294	523	34	1
9	2975	12379	12100	7045	2237	297	16	0
10	2639	10180	9491	5281	1612	199	6	0
11	2360	8590	7664	4051	1167	128	6	0

dlzka intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice	osmice	devatice
12	2057	7179	6279	3304	884	88	2	0
13	1791	6221	5378	2593	688	58	4	0
14	1692	5339	4517	2127	511	35	1	0
15	1566	4534	3842	1753	415	35	1	0
16	1416	4040	3224	1512	340	29	1	0
17	1259	3560	2831	1226	282	11	0	0
18	1204	3282	2515	1073	219	14	0	0
19	1128	2936	2181	977	179	9	0	0
20	1072	2645	1914	818	142	9	0	0
21	916	2458	1669	675	113	7	0	0
22	991	2150	1564	599	92	3	0	0
23	848	2014	1397	559	75	5	0	0
24	793	1826	1348	470	62	1	0	0
25	771	1647	1131	434	64	3	0	0
26	766	1533	1036	373	38	3	0	0
27	670	1424	958	330	43	2	0	0
28	736	1352	877	323	32	1	0	0
29	620	1225	783	306	41	0	0	0
30	589	1122	729	271	32	0	0	0
31	623	1068	689	247	22	0	0	0
32	535	971	640	201	21	1	0	0
33	507	949	586	186	11	0	0	0
34	500	804	529	186	18	0	0	0
35	488	812	490	169	15	0	0	0
36	470	779	496	138	14	2	0	0
37	461	769	419	126	14	2	0	0
38	404	720	402	109	7	0	0	0
39	420	656	378	94	6	0	0	0
40	374	625	380	92	6	0	0	0

dlzka intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice	osmice	devatice
41	393	611	326	81	6	0	0	0
42	360	605	343	90	10	0	0	0
43	375	499	313	83	8	0	0	0
44	346	563	275	81	3	0	0	0
45	332	476	261	75	1	0	0	0
46	322	484	290	59	2	0	0	0
47	326	432	266	69	2	0	0	0
48	305	419	236	63	1	0	0	0
49	299	408	205	53	3	0	0	0
50	293	425	197	44	2	0	0	0
51	266	388	196	39	1	0	0	0
52	274	406	212	37	5	0	0	0
53	247	311	207	41	1	0	0	0
54	243	332	176	27	3	0	0	0
55	241	309	170	42	1	0	0	0
56	244	280	153	27	1	0	0	0
57	233	265	154	36	1	0	0	0
58	213	301	135	31	1	0	0	0
59	234	283	131	24	0	0	0	0
60	219	276	136	20	1	0	0	0
61	200	266	139	23	2	0	0	0
62	213	253	120	17	1	0	0	0
63	223	228	114	20	4	0	0	0
64	184	244	128	21	0	0	0	0
65	190	227	99	21	0	0	0	0
66	167	205	112	14	0	0	0	0
67	185	213	92	16	1	0	0	0
68	183	195	86	15	0	0	0	0
69	150	188	98	16	0	0	0	0

dlzka intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice	osmice	devatice
70	181	200	81	16	1	0	0	0
71	159	192	84	18	0	0	0	0
72	152	177	58	12	0	0	0	0
73	139	173	72	9	1	0	0	0
74	179	154	57	7	0	0	0	0
75	150	157	75	7	0	0	0	0
76	133	175	63	11	0	0	0	0
77	125	157	70	9	0	0	0	0
78	142	171	59	6	0	0	0	0
79	130	135	67	4	0	0	0	0
80	138	129	60	11	1	0	0	0
81	144	134	73	7	0	0	0	0
82	127	148	77	7	0	0	0	0
83	118	116	47	5	0	0	0	0
84	132	111	66	5	1	0	0	0
85	111	136	47	7	0	0	0	0
86	117	103	58	3	1	0	0	0
87	111	123	51	1	0	0	0	0
88	114	103	40	6	0	0	0	0
89	109	103	41	3	0	0	0	0
90	122	114	47	5	0	0	0	0
91	113	122	53	4	0	0	0	0
92	110	107	39	1	0	0	0	0
93	111	103	35	2	0	0	0	0
94	91	106	52	4	1	0	0	0
95	112	91	52	4	0	0	0	0
96	104	81	40	4	0	0	0	0
97	85	79	32	1	0	0	0	0
98	79	100	38	0	0	0	0	0



dlzka intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice	osmice	devatice
99	109	96	27	6	0	0	0	0
100	86	80	35	2	0	0	0	0
101	82	104	30	1	0	0	0	0
102	97	86	31	3	0	0	0	0
103	83	75	32	5	0	0	0	0
104	91	69	23	2	0	0	0	0
105	87	71	23	0	0	0	0	0
106	85	72	32	3	0	0	0	0
107	78	96	35	0	0	0	0	0
108	84	66	31	1	0	0	0	0
109	87	73	26	2	0	0	0	0
110+	8630	4873	978	47	0	0	0	0

Tabuľka 3: Lekárne 2016 nad 15 km

dlzka_intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice
6	2838	2696	1277	425	65	3
7	2278	1806	871	288	34	0
8	1710	1417	557	159	17	0
9	1361	994	431	110	7	0
10	1126	743	324	59	2	0
11	903	590	261	54	2	0
12	840	521	195	25	2	0
13	643	417	162	37	1	0
14	572	340	123	11	0	0
15	483	289	106	13	1	0
16	477	257	95	13	0	0
17	419	192	60	6	0	0
18	369	199	56	5	0	0
19	294	170	46	5	0	0

dlzka_intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice
20	335	153	35	3	0	0
21	281	144	30	1	0	0
22	257	101	34	2	0	0
23	210	85	20	2	0	0
24	210	100	23	1	0	0
25	199	80	20	0	0	0
26	179	89	12	0	0	0
27	161	67	13	0	0	0
28	135	68	13	1	0	0
29	156	63	9	0	0	0
30	133	46	10	0	0	0
31	103	48	12	0	0	0
32	131	51	8	0	0	0
33	105	37	3	0	0	0
34	102	41	6	0	0	0
35	97	35	6	0	0	0
36	88	30	5	0	0	0
37	89	29	8	0	0	0
38	72	26	5	0	0	0
39	55	35	4	1	0	0
40	68	26	1	0	0	0
41	62	25	1	0	0	0
42	54	17	4	0	0	0
43	58	23	1	0	0	0
44	60	12	2	0	0	0
45	66	13	3	0	0	0
46	58	9	1	1	0	0
47	63	13	1	0	0	0
48	44	13	0	0	0	0
49	42	13	3	0	0	0
50	45	13	1	0	0	0

dlzka_intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice
51	45	13	1	0	0	0
52	33	15	3	0	0	0
53	41	16	1	0	0	0
54	40	8	1	0	0	0
55	46	9	2	1	0	0
56	37	9	0	0	0	0
57	15	12	2	0	0	0
58	31	4	0	0	0	0
59	24	5	2	0	0	0
60	29	15	3	0	0	0
61	25	6	0	0	0	0
62	37	11	0	0	0	0
63	21	5	0	0	0	0
64	18	9	0	0	0	0
65	22	9	0	0	0	0
66	22	7	0	0	0	0
67	23	6	0	0	0	0
68	26	3	0	0	0	0
69	24	4	0	0	0	0
70	20	3	0	0	0	0
71	11	7	0	0	0	0
72	18	5	0	0	0	0
73	16	4	0	0	0	0
74	14	6	0	0	0	0
75	15	5	0	0	0	0
76	21	3	0	0	0	0
77	21	5	0	0	0	0
78	17	2	0	0	0	0
79	13	2	0	0	0	0
80	15	3	0	0	0	0
81	13	4	0	0	0	0

dlzka_intersectu	dvojice	trojice	stvorice	patice	sestice	sedmice
82	10	3	0	0	0	0
83	10	3	0	0	0	0
84	13	2	0	0	0	0
85	12	1	0	0	0	0
86	12	4	0	0	0	0
87	10	1	0	0	0	0
88	14	2	0	0	0	0
89	11	4	0	0	0	0
90	10	1	0	0	0	0
91	16	1	0	0	0	0
92	10	3	0	0	0	0
93	11	1	0	0	0	0
94	10	0	0	0	0	0
95	15	3	0	0	0	0
96	8	3	0	0	0	0
97	4	0	1	0	0	0
98	11	1	0	0	0	0
99	6	5	1	0	0	0
100	4	1	0	0	0	0
101	9	1	0	0	0	0
102	13	4	0	0	0	0
103	8	2	0	0	0	0
104	4	1	0	0	0	0
105	6	4	0	0	0	0
106	5	0	1	0	0	0
107	10	1	0	0	0	0
108	5	2	0	0	0	0
109	7	1	0	0	0	0
110+	452	33	1	0	0	0

Tabuľka 4: Pômocky

dlzka_intersectu	dvojice	trojice	stvorice	patice	sestice
3	3523	2582	578	89	8
4	2093	1133	194	28	1
5	1343	580	83	11	0
6	969	343	54	4	0
7	737	244	29	3	0
8	553	156	15	0	0
9	445	113	21	1	0
10	393	92	13	0	0
11	315	62	9	0	0
12	264	59	5	0	0
13	228	24	6	0	0
14	162	34	5	0	0
15	161	22	1	0	0
16	124	28	2	0	0
17	117	21	1	0	0
18	106	17	1	0	0
19	107	12	0	0	0
20	77	10	1	0	0
21	81	8	0	0	0
22	63	11	0	0	0
23	69	10	0	0	0
24	56	7	0	0	0
25	57	8	1	0	0
26	51	7	0	0	0
27	56	5	2	0	0
28	35	3	0	0	0
29	37	5	0	0	0
30	44	5	0	0	0
31	26	5	0	0	0
32	32	3	0	0	0

dlzka_intersectu	dvojice	trojice	stvorice	patice	sestice
33	29	4	1	0	0
34	26	2	0	0	0
35	26	3	0	0	0
36	29	3	0	0	0
37	20	2	0	0	0
38	23	0	0	0	0
39	18	2	0	0	0
40	11	0	0	0	0
41	13	0	0	0	0
42	17	0	0	0	0
43	15	5	0	0	0
44	13	0	0	0	0
45	9	0	0	0	0
46	9	0	0	0	0
47	10	0	0	0	0
48	16	1	1	0	0
49	14	0	0	0	0
50	10	0	0	0	0
51	12	1	0	0	0
52	9	0	0	0	0
53	7	0	0	0	0
54	2	1	0	0	0
55	8	0	0	0	0
56	5	0	0	0	0
57	9	1	0	0	0
58	3	0	0	0	0
59	6	0	0	0	0
60	5	0	0	0	0
61	2	0	0	0	0
62	7	0	0	0	0

dlzka_intersectu	dvojice	trojice	stvorice	patice	sestice
63	8	0	0	0	0
64	8	0	0	0	0
65	3	0	0	0	0
66	4	0	0	0	0
67	2	0	0	0	0
68	5	0	0	0	0
69	5	0	0	0	0
70	4	0	0	0	0
71	1	0	0	0	0
72	1	0	0	0	0
73	2	0	0	0	0
74	4	0	0	0	0
75	3	0	0	0	0
76	0	1	0	0	0
77	1	0	0	0	0
78	1	0	0	0	0
79	4	0	0	0	0
80	1	0	0	0	0
81	2	0	0	0	0
82	5	0	0	0	0
83	5	0	0	0	0
84	5	2	0	0	0
85	1	0	0	0	0
86	5	0	0	0	0
87	1	0	0	0	0
89	2	0	0	0	0
90	2	0	0	0	0
91	3	0	0	0	0
92	5	0	0	0	0
93	2	0	0	0	0

dlzka_intersectu	dvojice	trojice	stvorice	patice	sestice
94	2	0	0	0	0
96	5	0	0	0	0
97	2	0	0	0	0
98	3	0	0	0	0
100+	61	4	0	0	0

Tabuľka 5: ŠAS na KOD NZ

dlzka_intersectu	dvojice	trojice	stvorice
101	86	9	0
102	89	10	0
103	84	5	0
104	114	12	0
105	74	9	1
106	66	7	0
107	72	6	0
108	81	5	0
109	70	7	0
110	75	7	0
111	85	4	0
112	62	2	0
113	65	4	0
114	62	4	0
115	67	4	0
116	58	4	0
117	65	3	0
118	72	2	0
119	54	7	0
120	57	9	1
121	51	2	0



dlzka_intersectu	dvojice	trojice	stvorice
122	61	6	0
123	61	3	0
124	45	2	0
125	46	3	0
126	47	8	0
127	54	5	0
128	49	5	0
129	43	4	0
130	41	8	0
131	48	3	0
132	55	2	0
133	48	1	0
134	36	4	0
135	54	0	0
136	45	4	0
137	36	1	0
138	33	3	0
139	27	4	0
140	41	3	0
141	28	1	0
142	37	3	0
143	38	1	0
144	31	1	0
145	33	4	0
146	40	6	0
147	32	7	0
148	23	0	0
149	17	3	0
150	28	2	0
151	27	2	0
152	27	4	0

dlzka_intersectu	dvojice	trojice	stvorice
153	30	1	0
154	38	1	0
155	30	4	0
156	34	5	0
157	31	2	0
158	27	3	0
159	23	1	0
160	26	2	0
161	25	1	0
162	20	3	0
163	24	5	0
164	24	0	0
165	22	0	0
166	26	1	0
167	22	1	0
168	29	2	0
169	33	0	0
170	13	1	0
171	17	2	0
172	19	2	0
173	21	0	0
174	19	2	0
175	27	1	0
176	19	1	0
177	24	0	0
178	25	1	0
179	27	0	0
180	25	1	0
181	24	1	0
182	26	0	0
183	17	2	0

dlzka_intersectu	dvojice	trojice	stvorice
184	21	2	0
185	14	0	0
186	10	1	0
187	13	0	0
188	19	1	0
189	16	0	0
190	16	2	0
191	15	1	0
192	21	0	0
193	19	1	0
194	14	1	0
195	13	0	0
196	10	0	0
197	22	1	0
198	15	1	0
199	21	2	0
200+	2655	45	0

Tabuľka 6: ŠAS na KÓD VZ a ODB

dlzka_intersectu	dvojice	trojice	stvorice
101	114	13	0
102	141	6	0
103	105	6	0
104	122	3	0
105	115	7	0
106	91	3	0
107	121	5	0
108	108	3	0
109	102	2	0
110	99	3	0

dlzka intersectu	dvojice	trojice	stvorice
111	83	10	0
112	105	5	0
113	90	3	0
114	88	6	0
115	79	3	0
116	74	0	0
117	90	2	0
118	81	4	0
119	65	4	0
120	78	2	0
121	80	4	0
122	67	1	0
123	73	1	0
124	86	4	0
125	53	2	0
126	60	5	0
127	68	2	0
128	77	6	0
129	64	6	0
130	68	6	0
131	69	5	0
132	85	0	0
133	65	0	0
134	41	5	0
135	57	1	0
136	61	3	0
137	59	4	0
138	50	2	0
139	39	5	0
140	49	3	0
141	44	2	0

dlzka intersectu	dvojice	trojice	stvorice
142	56	3	0
143	49	2	0
144	44	4	0
145	53	0	0
146	48	3	0
147	53	2	0
148	46	1	0
149	44	3	0
150	42	4	0
151	51	2	0
152	32	3	0
153	45	0	0
154	44	1	1
155	40	5	0
156	36	0	0
157	35	0	0
158	40	0	0
159	40	0	0
160	33	1	0
161	42	1	0
162	36	2	0
163	30	1	0
164	29	2	0
165	37	3	0
166	32	1	0
167	33	0	0
168	27	2	0
169	38	0	0
170	33	2	0
171	29	0	0
172	27	2	0

dlzka intersectu	dvojice	trojice	stvorice
173	36	1	0
174	29	2	0
175	39	2	0
176	29	0	0
177	29	0	0
178	32	1	0
179	31	0	0
180	30	0	0
181	28	2	0
182	31	0	0
183	28	1	0
184	20	0	0
185	23	1	0
186	25	1	0
187	27	0	0
188	25	2	0
189	21	3	0
190	31	1	0
191	30	1	0
192	22	2	0
193	28	0	0
194	21	1	0
195	25	0	0
196	26	0	0
197	19	0	0
198	23	1	0
199	29	0	0
200+	1922	19	0