**ČASŤ 1: MAGICKÉ ŠTVORCE A ŠTVORCE**

**Definícia 1.1**: Maticu veľkosti 3x3

nazývame **magickým štvorcom**, ak obsahuje navzájom rôzne kladné celé čísla a zároveň majú výrazy

rovnakú hodnotu rovnú S. Hodnotu S nazývame **súčet magického štvorca**.

**Veta 1.2**: Pre každý magický štvorec platí .

**Dôkaz**: Vyplýva priamo zo vzťahu.

**Dôsledok 1.2**: Pre každý magický štvorec platí .

**Veta 1.3**: Pre každý magický štvorec platí

**Dôkaz**: Dokážeme iba (i), postup dôkazu pri (ii), (iii) a (iv) je analogický.

Platí a zároveň . Z prvého vzťahu postupne vyjadríme a . Z druhého vzťahu najprv vyjadríme . Dosadím do rovnice dostaneme

čo po úprave vyjde

Keďže , platí aj (čo bolo treba dokázať).

**Definícia 1.4**: Celé číslo sa nazýva **štvorec** ak sa dá napísať v tvare , kde je celé číslo.

**ČASŤ 2: ALGEBRAICKÉ IDENTITY**

**Lema 2.1**: Celé čísla sa dajú zapísať ako (kde sú tiež celé čísla) práve vtedy, keď sú čísla obe párne alebo obe nepárne.

**Dôkaz**: Sústava rovníc

má riešenie . Ak sú čísla obe párne alebo obe nepárne, tak ich súčet aj rozdiel je deliteľný dvomi, teda sú celé čísla. Ak je číslo párne a číslo nepárne (opačný prípad je analogický), ich súčet aj rozdiel je nepárny, teda nemôžu byť celé čísla.

**Lema 2.2**: Nech je štvorec. Potom

1. Ak je párne, tak je deliteľné .
2. Ak je nepárne, tak dáva po delení zvyšok .

**Dôkaz**: Ak je štvorec, tak sa z definície dá napísať ako , kde je celé číslo.

1. Ak je párne, tak aj je párne. To znamená, že existuje celé číslo také, že . Tým dostávame , takže je deliteľné .
2. Ak je nepárne, tak aj je nepárne. To znamená, že existuje celé číslo také, že . Tým dostávame , takže dáva po delení zvyšok .

**Veta 2.3**: Množina všetkých celočíselných riešení rovnice je daná parametricky:

kde sú celé čísla, pre ktoré platí vzťah .

**Dôkaz**: Je zrejmé, že daná parametrizácia vygeneruje vždy správne riešenie – dosadením dostaneme

z čoho automaticky vyplýva .

Ukážeme aj to, že každé riešenie sa dá takto parametricky vyjadriť. Vidíme, že členy na ľavej a pravej strane rovnice vieme popárovať do dvojíc , kde sú celé čísla. Podľa Lemy 2.1 je to možné práve vtedy, keď sú oba členy párne alebo nepárne.

Stačí preto dokázať, že v danej rovnici musí byť na ľavej aj pravej strane rovnaký počet párnych (a teda aj nepárnych) členov. Postupujme sporom. Predpokladajme, že počet párnych členov a nepárnych členov na jednej strane rovnice je iný ako na druhej strane. Máme dve možnosti:

* Na jednej strane rovnice sú len nepárne členy a na druhej strane sú dva párne a jeden nepárny. Potom však z Lemy 2.2 vyplýva, že jedna strana rovnice dáva po delení 4 zvyšok 3, zatiaľ čo druhá strana dáva po delení 4 zvyšok 1, čo je spor.
* Na jednej strane rovnice sú len párne členy a na druhej strane sú dva nepárne a jeden párny. Potom však z Lemy 2.2 vyplýva, že jedna strana rovnice je deliteľná 4, zatiaľ čo druhá strana dáva po delení 4 zvyšok 2, čo je spor.

**Veta 2.4**: Množina všetkých celočíselných riešení rovnice je daná parametricky:

kde sú celé čísla, pre ktoré platí a zároveň hodnota je celé číslo.

**Dôkaz**: Použitím vety 2.3 na rovnicu dostaneme parametrizáciu:

Z posledného vzťahu vyjadríme , . Keďže z vety 2.3 musí platiť aj vzťah , dostaneme:

Premennú môžeme takto vyjadriť len za podmienky a zlomok musí byť celé číslo - k tomu sa vrátime na konci dôkazu. Z toho zároveň vyplýva, že:

Dosadíme do pôvodných premenných :

Substitúciou dostaneme hľadaný vzťah. Skontrolujeme ešte podmienky pri vyjadrení .

* Podmienka je ekvivalentná s  - to platí zo zadania.
* je celé číslo je ekvivalentné s tým, že je celé číslo - to tiež platí zo zadania.

**Veta 2.5**: Množina všetkých celočíselných riešení rovnice je daná parametricky:

kde .

**Veta 2.6**: Množina všetkých celočíselných riešení rovnice je daná parametricky:

kde .

**Dôkaz**: Použitím vety 2.5 na rovnicu dostaneme parametrizáciu:

Z posledného vzťahu vyjadríme . To dosadíme za :

Výraz môžeme beztrestne považovať ako násobok nejakého riešenia (vieme ho regulovať premennou ). Ak vo zvyšku použijeme substitúciu , dostaneme hľadaný výraz.

**Veta 2.7**: Množina všetkých celočíselných riešení rovnice je daná parametricky:

kde .

**Dôkaz**: Rovnicu riešime ako kvadratickú s premennou :

Keďže , aj . Preto nutne pre nejaké . To znamená, že:

Rovnicu riešime ako kvadratickú s premennou :

Keďže , aj . Preto nutne pre nejaké . To znamená, že

Dostali sme sa k rovnici , ktorá má podľa vety 2.6 parametrizáciu:

Podľa toho platí:

To znamená, že:

Substitúciou dostaneme hľadaný vzťah.

**ČASŤ 3: CIEĽ ROČNÍKOVÉHO PROJKETU**

**Hypotéza 7**: Jediný magický štvorec, pre ktorý platí, že 7 z jeho 9 prvkov sú štvorce, je

a všetky jeho rotácie, symetrie, alebo násobky.

**Hypotéza 8**: Neexistuje magický štvorec, pre ktorý platí, že 8 z jeho 9 prvkov sú štvorce.

**Hypotéza 9**: Neexistuje magický štvorec, pre ktorý platí, že každý jeho prvok je štvorec.

Našou úlohou bude vytvoriť kolekciu algoritmov, ktoré budú efektívne preverovať hypotézu 7. Použijeme na to jazyk Python,

**ČASŤ 4: VETY PRE HYPOTÉZU 7**

**Veta 4.1**: Ak existuje magický štvorec v tvare

tak musia existovať také, že:

**Dôkaz**: Z definície magického štvorca a vety 1.3 musia platiť nasledovné vzťahy:

Aplikovaním vety 2.5 na (i) a vety 2.6 na (ii) a (iii) dostaneme:

Sčítaním vzťahov pre dostaneme , ich odčítaním zase .

Sčítaním vzťahov pre dostaneme , ich odčítaním zase .

Zo vzťahu vyjadríme a zo vzťahu vyjadríme .

Dosadením do vzťahu dostaneme , z čoho .

Dosadením do posledného vzťahu dostaneme:

z čoho vieme odvodiť . To je ekvivalentné s

**Veta 4.2**: Ak existuje magický štvorec v tvare

tak musia existovať navzájom rôzne celé čísla také, že platí:

1. ani jeden z výrazov , , , , nie je rovný nule

**Dôkaz**: Pre daný magický štvorec musí platiť:

Na prvú sústavu rovníc dvakrát uplatníme vetu 2.4. Tým dostaneme:

Túto parametrizáciu na tri krát dosadíme do druhej sústavy rovníc:

Keď z parametrizácie využijeme, že , dostaneme po úprave rovnosť

Analogickým postupom dostaneme po dosadení do rovníc , resp. rovnosti, resp. . Po spojení všetkých troch rovností s parametrizáciou získame hľadanú rovnosť.

**Veta 4.3**: Ak existuje magický štvorec v tvare

tak musia existovať také, že:

**Dôkaz**: Pre daný magický štvorec musí platiť:

Aplikovaním vety 2.6 na prvú sústavu rovníc dostaneme vzťahy:

Zo vzťahu pre si vyjadríme a .

Dosadením do druhej rovnice dostaneme:

To vieme ekvivalentnými úpravami upraviť na tvar:

Keď v tomto tvare dosadíme a a potom sa zbavíme menovateľov, dostaneme rovnosť zo zadania.

**Veta 4.4**: Ak existuje magický štvorec v tvare

tak musia existovať také, že výrazy

majú rovnakú hodnotu.

**Dôkaz**: Pre daný magický štvorec musí platiť:

Aplikovaním vety 2.6 na všetky sústavy rovníc dostaneme vzťahy:

Zo vzťahov pre vyjadríme , , . Po dosadení týchto vyjadrení do vzťahu pre a ekvivalentných úpravách dostaneme vzťah zo zadania.

**Veta 4.5**: Magické štvorce v jednom z nasledovných tvarov

alebo alebo

vieme parametrizovať tak, že a zvyšné štyri členy sú v tvare:

**Dôkaz**: Dôkaz urobíme len pre prvý tvar magického štvorca, ostatné tvary sa dokazujú analogicky.

Musí platiť . Použitím vety 2.6 dostaneme vzťahy:

Zo vzťahu pre vyjadríme . Potom zvolíme substitúciu (môžeme, lebo všetky premenné teraz vieme regulovať pomocou ). Tým dostaneme parametrizáciu zo zadania.

**ČASŤ 5: ALGORITMY NA HYPOTÉZU 7**

**Algoritmy pre vetu 4.1**:

* Nájdeme celé také, že . Potom nájdeme také celé , aby boli zlomky celé čísla. To urobíme tak, že vypočítame a ak je deliteľný výrazom , tak definujeme . Z neho vypočítame . Potom vypočítame členy potenciálneho magického štvorca:

Šesť členov sú určite štvorce, ostáva overiť magickosť a to, či nejaké z  sú štvorce.

**Algoritmy pre vetu 4.2:**

* Generujeme všetky možné podľa vety 4.1 a overujeme (i) a (ii).
* Generujeme všetky možné podľa vety 2.7. Potom . Snažíme sa pre rôzne získať rovnaké , čím získame odlišné , pre ktoré platí . Tieto dvojice budú postupne dvojicami . Tým máme z vety 4.2 splnených viac rovníc a musíme overovať len podmienku .
* Pred generovaním urobíme šikovné pozorovanie. Všimnime si, že ak platí z vety 4.2 , tak platí aj (to nie je ťažké dokázať). Preto generujeme len rôzne dvojice pre dané . Potom stačí nájsť také , že platí .
* Ešte šikovnejší trik je prepísať si sústavu z vety 4.2 nasledovne:

Potom po vhodnej substitúcii riešime sústavu rovníc v obore racionálnych čísel.

**Algoritmy pre vetu 4.3**: Generujeme všetky možné podľa vety 4.3.

**Algoritmy pre vetu 4.4**: Generujeme všetky možné podľa vety 4.4.

**Algoritmy pre vetu 4.5:** Uvedené v nasledovnej časti.

**ČASŤ 6: IMPLEMENTÁCIA PROGRAMU**

Počas zimného a letného semestra sme sa zamerali na efektívnu implementáciu programu. Hľadali sme algoritmus pre vetu 4.5. Spomedzi všetkých viet sme si zvolili práve tú, lebo

1. bola flexibilná a ľahko implementovateľná
2. zaručovala, že každý skúmaný magický štvorec obsahuje aspoň 5 štvorcov (silná vlastnosť)
3. výsledok sa dal dobre graficky reprezentovať (t. j. koľko štvorcov sa našlo pre akú hodnotu)

Využitím iných viet by sme si prácu zbytočne skomplikovali.

Program sa spúšťa cez príkazový riadok. Používateľ zadá päť parametrov:

* typ vyhľadávania

1. úplné (bruteforce) – prehľadá všetky v danom intervale
2. náhodné (random) – prehľadá náhodné v intervale daný počet krát

* dolnú hranicu intervalu
* hornú hranicu intervalu
* počet iterácií v náhodnom prehľadávaní (dobrovoľné)
* počet využitých CPU jadier pri prehľadávaní (dobrovoľné)

Počas behu algoritmu sa používateľovi priebežne vypisuje progress bar a údaj o tom, koľko (približne) času ostáva do konca. Všetky nájdené riešenia sa zapisujú do textového súboru. Na konci sa ukáže graf početnosti vyhovujúcich magických štvorcov (tých, čo majú aspoň 6 štvorcov) na danom intervale. Ak sa našli aj riešenia so 7 štvorcami, vypíšu sa.

Pri vymýšľaní algoritmu sme stáli pred neľahkou úlohou: ako prechádzať vyhovujúce čo najefektívnejšie (parameter z vety slúži len ako konštantný násobok, ktorý na konečný výsledok nemá vplyv). Parametre vedeli vytvoriť túto množinu štvorcov:

Môžeme si všimnúť nasledovné štyri veci:

* Ak (resp. ), tak štvorce nevyhovujú, pretože obsahujú duplicitné prvky.
* Ak vymeníme medzi sebou a (resp. a ), nič nepokazíme – dostaneme rovnakú množinu možných štvorcov (akurát osovo prevrátených).
* Ak vymeníme medzi sebou a a zároveň a , nič nepokazíme – dostaneme rovnakú množinu možných štvorcov (akurát osovo prevrátených).
* Ak majú a (resp. a ) spoločného deliteľa väčšieho ako 1, tento deliteľ sa prejaví ako konštantný násobok a nebude mať vplyv na výsledok.

Na základe týchto vlastností stačí skúmať také , že

Aby sme prechádzali naozaj len potrebné , vytvoríme si pole, do ktorého pridáme všetky usporiadané dvojice čísel také, že a zároveň . Toto pole na konci utriedime podľa hodnoty v každej dvojici. O všetko sa stará funkcia **add\_all\_pairs**.

Pri vyhľadávaní potom funkcia **run\_all\_pairs** (resp. **run\_random\_pairs**) prejde všetky (resp. náhodné) páry týchto dvojíc a . Pre každý vygeneruje funkcia **run\_configuration** množinu štvorcov (tie, čo sú vypísané vyššie). Každý štvorec je najprv vykrátený a potom umocnený na druhú funkciou **normalize**. Ak sa ukáže, že štvorec neobsahuje duplicitné prvky a obsahuje aspoň šesť štvorcov, je na konci výpočtu pridaný medzi riešenia (vyhovujúce štvorce) funkciou **add\_solution**.

Ostal nám problém: ako si pamätať, či sme dané riešenie už predtým nenašli? Ak nejaký štvorec vznikne rotáciou, symetriou alebo násobkom z iného nájdeného štvorca, tak sme nové riešenie nenašli. Pomôže nám nasledovná veta:

**Veta 6.1**: Každý magický štvorec je jednoznačne definovaný svojím stredom a dvoma najmenšími rohmi, pričom tieto tri hodnoty sú nesúdeliteľné.

**Dôkaz**: Vezmeme si ľubovoľný magický štvorec a ukážeme si, ako ho dať do spomínaného tvaru. Ak sú jeho hodnoty súdeliteľné, jednoducho ich vykrátime. Tým máme jednoznačne určený stred . Teraz budeme štvorec otáčať dovtedy, kým sa jeho najmenší roh (označíme ho ) nedostane do ľavého horného rohu. Označíme druhý najmenší roh . Teoreticky sú možné len tieto tri situácie:

alebo alebo

Dokážeme, že vždy vieme dostať do pravého horného rohu. Prvý prípad triviálne vyhovuje, zatiaľ čo v druhom prípade môžeme štvorec osovo prevrátiť podľa vedľajšej diagonály (tej, kde je ) a dostať na požadované miesto.

Tretí prípad nemôže nastať. Postupujme sporom. Nech je magický štvorec, ktorý má stred a jeho dva najmenšie rohy sú Na opačnej diagonále sa však potom nachádzajú dva najväčšie rohy, z čoho vyplýva, že súčet hodnôt na diagonálach je rôzny (to je spor s magickosťou).

Z troch hodnôt vieme jednoznačne zrekonštruovať magický štvorec. Magický súčet je 3S (podľa vety 1.2), čím vieme dopočítať zvyšné dva rohy. Z toho už dopočítame aj ostatné hodnoty.

Ak si chceme zapamätať magický štvorec, stačí si uložiť len hodnoty (napr. do poľa, alebo ako reťazec do asociatívneho poľa - ako je implementované v našom programe). Tento reťazec sme si pomenovali **index magického štvorca** (v programe ho vracia funkcia **get\_index**).

**ČASŤ 7: VÝSLEDKY**

V zimnom semestri sme sa zamerali na vetu 4.5. Konkrétne sme generovali všetky magické štvorce, ktoré obsahovali aspoň 5 štvorcov. Využili sme všetky tri konfigurácie z vety 4.5:

alebo alebo

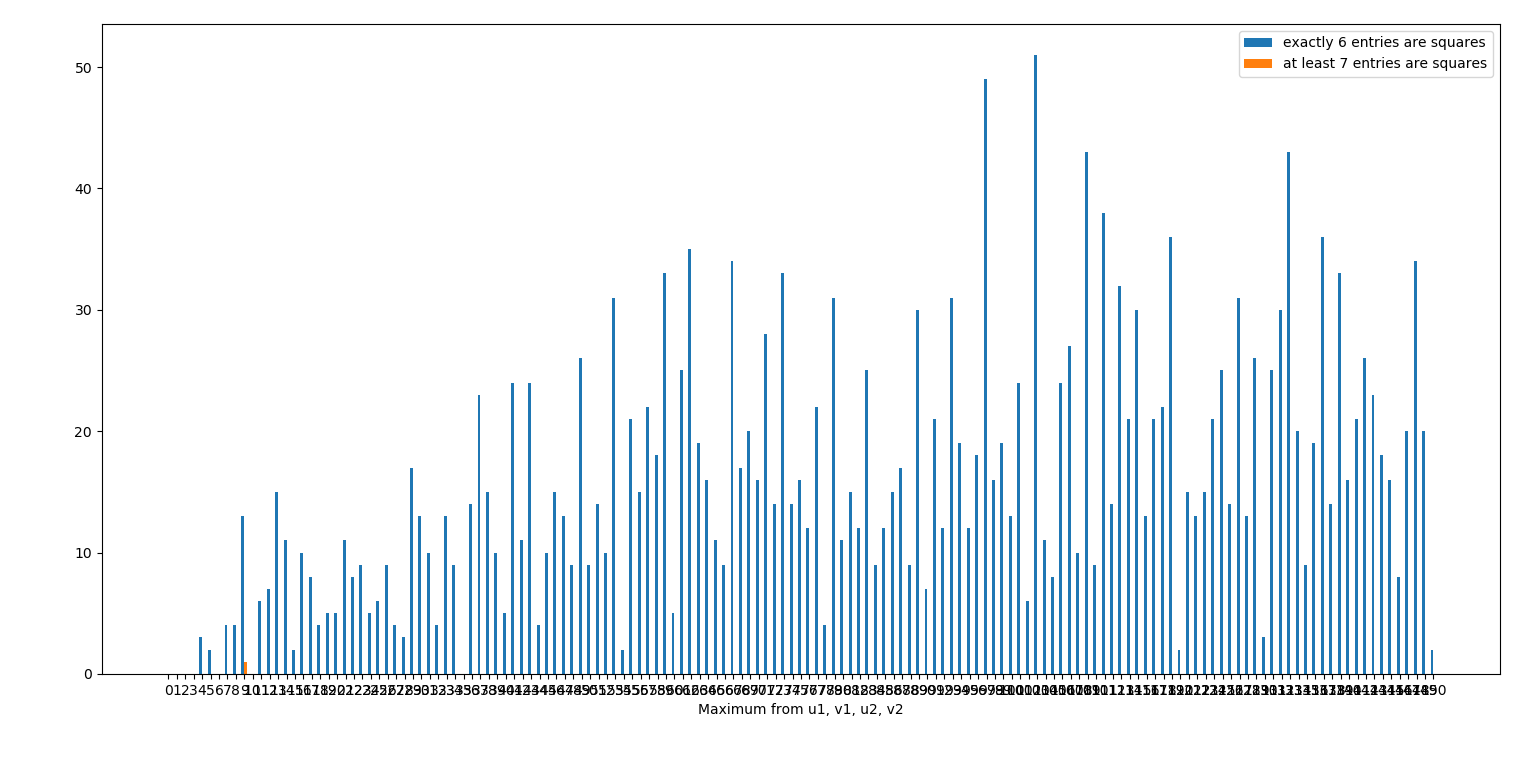
V programe je možné generovať riešenia náhodne alebo hrubou silou. Na konci výpočtu hrubou silou sa objaví graf. Ten popisuje, koľko rôznych magických štvorcov s aspoň šiestimi štvorcami existuje pre dané maximum zo štvorice .

Graf riešení pre :

Obrázok, na ktorom je objekt, anténa

Automaticky generovaný popis

Graf riešení pre :



Pre teda existuje iba jediné riešenie so siedmimi štvorcami (to, ktoré poznáme).

V letnom semestri sme pokračovali s vetou 4.5. Náš program prešiel výraznými zmenami:

* Odstránili sme chybu pri počítaní, ktorá spôsobila, že sme tú istú konfiguráciu overovali viac krát. To výrazne zredukovalo počet nutných operácií.
* Pridali sme možnosť výberu spodnej hranice. Odteraz je možné hľadať riešenia v ľubovoľnom intervale od do ( je spodná hranica, je vrchná hranica). Ak napr. úspešne prehľadáme všetko v intervale od do , už sa stačí zaoberať len riešeniami väčšími ako .
* Parametre sa zadávajú cez argparser, nie cez štandardný vstup.
* Komplikovaný pytest nahradil lepší unittest.
* Zvýšili sme odolnosť programu voči výpadkom. Po každej iterácii sa zapíšu do logu (textového súboru) nájdené magické štvorce (tie, pre ktoré platí, že aspoň 6 ich čísel sú štvorce). Zápisy sú permanentné, čo znamená, že ak dôjde k výpadku, získané dáta sa nestratia.
* Odhad času, ktorý ostáva do konca bruteforce vyhľadávania, je omnoho presnejší. Ráta sa totiž vzhľadom na aktuálny výkon počítača.
* Najväčším zlepšením bola úspešná a efektívna implementácia multiprocessingu (paralelného programovania). Použili sme na to multiprocessing Pool a jeho metódu map (viac konfigurácií sa prechádzalo naraz). Tým sme výrazne znížili čas potrebný na prehľadávanie. Na demonštráciu: predtým trvalo prehľadávanie od do približne 7 hodín (teraz to zaberie menej ako 7 minút).

Od 28. 5. 2020 do 29. 5. 2020 prebehlo intenzívne prehľadávanie pre od do . Program na 8 CPU jadrách bežal približne hodín. Bolo nájdených magických štvorcov, ktoré obsahovali aspoň štvorcov. Bohužiaľ, podarilo sa nájsť iba jediné riešenie obsahujúce aspoň štvorcov (to, ktoré poznáme).

Zhustený graf riešení pre :

Obrázok, na ktorom je čln, obrovské, sedenie, voda

Automaticky generovaný popis

Poznámky

* čas potrebný na prehľadanie sa môže líšiť v závislosti od počtu CPU jadier, výkonu a aplikácii bežiacich na pozadí počítača
* grafy z letného a zimného semestra sa líšia, pretože použili iný spôsob prehľadávania... implementácia z časti 6 sa viaže na metódu použitú v letnom semestri