

Proiectarea Algoritmilor - Tema 1

Data publicare: 5 martie 2014

Deadline: 23 martie 2014 - 23:55

Ioana Alexandra Antoche
Andrei Pârvu
Andrei Poenaru

1 Introducere

Ajunși în anul doi la facultate, studenții de la Automatică și Calculatoare au învățat două concepte noi și interesante: strategia greedy și programarea dinamică. După ce și-au format o părere despre cele două, le-a fost cerut să rezolve două probleme pentru ca noile cunoștințe să nu rămână nefolosite (și implicit să se uite).

2 Prima problemă - Sauron - 40 de puncte

După ce Sauron a pierdut inelul puterii acesta a decis să se retragă pentru o vreme în ascunzătoarea sa secretă pentru a plănuî un nou atac. Fără inelul puterii acesta trebuie acum să se bazeze doar pe cele $2 * N$ perechi de orci care au supraviețuit dezastrului. El are notat pe o piatră vulcanică (suficient de mare) pentru fiecare membru al perechii câte victime a realizat în ultimul război. În fiecare pereche, primul orc are rangul mai mare, deci el se numește Stăpân, iar celălalt orc se numește Sclav. Pentru a le ridica moralul slujitorilor săi, Sauron s-a gândit să își răsplătească supușii, plătind $2 * N$ salarii, câte unul pentru fiecare pereche. Valoarea unui salariu reprezintă numărul de victime ale Stăpânului sau ale Sclavului dintr-o echipă. Pentru că Sauron este lacom din fire el vrea să minimizeze pe cât posibil suma totală plătită, impunându-și însă câteva reguli:

- Dintre cele $2 * N$ perechi el trebuie să aleagă N Stăpâni și N Sclavi.
- Trebuie să plătească exact N salarii corespunzătoare Stăpânilor aleși și exact N salarii corespunzătoare Sclavilor aleși
- **din fiecare pereche trebuie plătit un singur orc - Stăpânul sau Sclavul**

Dintre cei $2 * N$ orci plătiți, Sauron va trebui să formeze N noi perechi astfel:

- se va grupa obligatoriu un Stăpân cu un Sclav
- pentru orice nouă pereche, indicele perechii inițiale din care este ales
- Sclavul trebuie să fie mai mic decât indicele perechii inițiale din care este ales Stăpânul său. Indicii sunt dați de ordinea în care apar cele $2 * N$ perechi pe piatra vulcanică.

Cerințe:

Sauron și-a dat seama că nu este un bun strateg atunci când a pierdut inelul puterii, iar acum își dă seama că nu e nici un bun programator. De aceea, vă roagă pe voi să îl ajutați să rezolve această problemă. Fiind foarte lacom, răsplata voastră va fi doar de 40 de puncte dacă reușiți să îi spuneți atât suma minimă pe care o va plăti în total, cât și dacă îi dați o regrupare corectă a celor N perechi de orci care se vor forma în final.

Date de intrare:

Pe prima linie a fișierului de intrare se va afla numărul de perechi de orci ($2 * N$), în ordinea care apar pe piatra vulcanică. Pe următoarele $2 * N$ linii se vor afla câte două numere pe linie, reprezentând numărul de victime ale primului orc (Stăpânului) și numărul de victime ale celui de-al doilea orc (Sclavul).

Date de ieşire:

Pe prima linie din fişierul de ieşire veţi afişa suma minimă pe care poate Sauron să o plătească în total, iar pe următoarele N linii se vor afişa câte două valori: Stăpânul care va rămâne şi Sclavul cu care va fi împerecheat.

Exemplu:

date.in:

```
6
3 0
7 1
2 0
5 1
6 0
4 2
```

date.out

```
12
3 1
4 2
6 5
```

Explicaţie: $pt\ 2 * N = 6$

- id: stăpân sclav
- 1: 3 0
- 2: 7 1
- 3: 2 0
- 4: 5 1
- 5: 6 0
- 6: 4 2

Suma minimă care respectă condiţiile din enunţ se obţine plătind perechea 1 cu 0 galbeni, plătind perechea 2 cu 1 galben, plătind perechea 3 cu 2 galbeni, plătind perechea 4 cu 5 galben, plătind perechea 5 cu 0 galbeni şi plătind perechea 6 cu 4 galbeni.

Restricţii:

- $2 \leq 2 * N \leq 10000$
- se garantează că rezultatul problemei va încăpea pe un tip de date pe 32 de biţi cu semn.
- dacă calculaţi doar salariul minim, trebuie să existe pentru acesta o grupare validă (care să respecte şi a doua cerinţă), chiar dacă nu afişaţi cele N perechi.

- numărul de victime corespunzătoare Stăpânului poate fi mai mic decât numărul de victime corespunzătoare Sclavului

3 A doua problemă - Numărare cadre - 40 de puncte

În urma izbucnirii războiului dintre umanitate și mașinile inteligente, conducătorii oamenilor iau decizia de a acoperi cerul cu un nor dens și a împiedica pătrunderea luminii solare. Fiind privată de principala sursă de energie, societatea mașinilor intră într-o perioadă de recesiune, din care reușește să iasă prin exploatarea bioenergiei generate de oameni: o formă de energie similară celei emanate de un gamer înrăit atunci când se delectează cu jocul favorit, omorând inamici în stanga și în dreapta. Astfel, s-a ajuns la situația în care mulți oameni sunt captivi într-o realitate virtuală numită *the Matrix*. Denumirea provine de la modalitatea în care sunt transmise datele în interiorul sistemului: sub forma unor matrice de valori binare, având N linii și M coloane.

Faceți parte din divizia IT a rezistenței umane și credeți că ați descoperit o particularitate a cadrelor de date: pe orice coloană a matricei pare că nu se găsesc niciodată mai mult de K valori consecutive de 1. Pentru a vă testa teoria și a vă aduce contribuția la salvarea omenirii vă sunt necesare informații despre cadrele interceptate până în prezent. De asemenea, trebuie să determinați restul la împărțirea prin 40009 a numărului de cadre distincte având proprietatea dată.

Cerințe:

Aflați numărul de matrice binare cu N linii și M coloane care respectă proprietatea că pe orice coloană se află cel mult K valori de 1 consecutive.

Exemplu: Pentru $N = 3$, $M = 1$, $K = 2$ există 7 cadre distincte:

0	0	0	1	0	1	1
0	0	1	0	1	1	0
0	1	0	0	1	0	1

Date de intrare:

Fișierul de intrare pentru această problemă va conține pe prima și singura linie valorile N , M și K separate prin spațiu. Pentru exemplul de mai sus fișierul de intrare ar fi:

```
3 1 2
```

Date de ieșire:

Fișierul de ieșire conține pe prima linie numărul (*modulo* 40009) de cadre cu N linii, M coloane și care au cel mult K valori de 1 consecutive pe orice coloană. Pentru exemplul de mai sus fișierul de ieșire ar fi:

```
7
```

Restricții:

- $1 \leq N, M \leq 2.000.000.000$
- $0 \leq K \leq 40$

4 Bonus - 20 de puncte

Problema de bonus reprezintă o extensie a problemei 2.

După ce ați analizat suficiente date v-ați dat seama că proprietatea pe care o bănuiați nu este respectată. Nedorind să renunțați la teoria voastră, decideți să relaxați puțin restricțiile: din cele M coloane, M_1 conțin **cel mult** K valori de 1 consecutive, iar M_2 conțin **cel puțin** o secvență cu mai mult de K valori de 1 consecutive. În mod evident, $M = M_1 + M_2$.

Cerințe:

Aflați numărul de matrice binare cu N linii și $M_1 + M_2$ coloane care respectă proprietatea că pe M_1 coloane se află cel mult K valori de 1 consecutive, iar pe fiecare din celelalte M_2 coloane se găsește cel puțin câte o secvență având mai mult de K valori de 1 consecutive. Rezultatul va fi afișat *modulo* 40009.

Exemplu: Pentru $N = 3$, $M_1 = 1$, $M_2 = 1$, $K = 2$ există 14 cadre diferite:

0	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1
0	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1
0	1	1	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1

1	0	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1
1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0
1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	1

Date de intrare:

Fișierul de intrare pentru această problemă va conține pe prima și singura linie valorile N , M_1 , M_2 și K separate prin spațiu. Pentru exemplul de mai sus fișierul de intrare ar fi:

```
3 1 1 2
```

Date de ieșire:

Fișierul de ieșire conține pe prima linie numărul (*modulo* 40009) de cadre cu N linii, $M_1 + M_2$ coloane și care respectă noua proprietate. Pentru exemplul de mai sus fișierul de ieșire ar fi:

```
14
```

Restricții:

- $1 \leq N \leq 2.000.000.000$
- $0 \leq M_1, M_2 \leq 40.000$
- $1 \leq M_1 + M_2 \leq 40.000$
- $0 \leq K \leq 40$

5 Punctare

Punctajul pentru primele două probleme este de 40 de puncte fiecare, iar bonusul valorează 20 de puncte. 10 puncte vor fi acordate pentru coding-style, 10 puncte pentru comentarii și README. Pentru detalii puteți să vă uitați și peste regulile generale de trimitere a temelor.

Vor exista în total 20 de teste de 5 puncte fiecare, 8 teste pentru prima problemă, 8 teste pentru cea de-a doua și 4 teste pentru bonus. Pentru prima problemă, din cele 5 puncte ale unui test, 4 puncte se vor acorda pentru aflarea valorii corecte (prima linie din fișierul de ieșire) și 1 puncte pentru reconstituirea soluției - punctele pentru reconstituirea soluției sunt condiționate de aflarea valorii corecte. Pentru a doua problemă și pentru bonus, nu există punctaje parțiale - un test este fie trecut, fie nu.

Punctele pe teste sunt independente, punctajul pe un anumit test nefiind condiționat de alte teste.

Fiecare problemă va avea o limită de timp pe test (precizată mai jos) - dacă execuția programului pe un test al acelei probleme va dura mai mult decât limita de timp veți primi automat 0 puncte pe testul respectiv și execuția va fi întreruptă. În fișierul README va trebui să descrieți soluția pe care ați ales-o pentru fiecare problemă, să precizați complexitatea pentru fiecare și alte lucruri pe care le considerați util de menționat.

Corectorii își rezervă dreptul de a scădea puncte dacă vor considera acest lucru necesar.

6 Format arhiva și testare

Temele pot fi testate automat pe vmchecker - acesta suportă temele rezolvate în C/C++ și Java. Dacă doriți să realizați tema în alt limbaj trebuie să trimiteți un e-mail lui Traian Rebedea (traian.rebedea@cs.pub.ro) în care să îi cereți explicit acest lucru.

Arhiva cu rezolvarea temei trebuie să fie .zip și să conțină în rădăcina acesteia:

- Fișierul/fișierele sursă
- Fișierul Makefile
- Fișierul README

Fișierul pentru make trebuie denumit **obligatoriu** Makefile și trebuie să conțină următoarele reguli:

- build, care va compila sursele și va obține executabilele.
- run-p1, care va rula executabilul pentru problema 1
- run-p2, care va rula executabilul pentru problema 2
- run-p3, care va rula executabilul pentru problema bonus
- clean, care va șterge executabilele generate.

Atenție! Numele regulilor trebuie să fie exact cele de mai sus, în special pentru cele de run. Absența sau denumirea diferită a acestora va avea drept consecință obținerea a 0 puncte pe testele echivalente problemei rezolvate de regula respectivă.

Atenție Pentru cei ce folosesc C/C++ **NU** este permisă compilarea cu opțiuni de optimizare a codului (O1, O2, etc.).

Atenție Pentru fiecare problemă, fișierul de intrare se va numi *date.in*, iar cel de ieșire *date.out*.

Limitele de timp pentru problema 1 sunt:

- C/C++ 0.5 secunde
- Java 0.6 secunde

Limitele de timp pentru problema 2 si problema bonus sunt:

- C/C++ 0.5 secunde
- Java 0.5 secunde