

Proiectarea Algoritmilor - Tema 1

Data publicare: 2 martie 2013

Deadline: 20 martie 2013 - 23:55

Claudia Cârdei
Andrei Pârvu
Ștefan Rușeți

- 6.03.2013 - Schimbare formă Makefile și adăugare limite de timp.

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 - Eliminare gard - 40 de puncte

Ion și Vasile sunt vecini de multă vreme, dar în ultimul timp relațiile dintre ei s-au deteriorat. Casele celor 2 sunt separate printr-un gard de lungime L format din N stâlpi (numerotarea se face de la stânga de dreapta, de la stâlpul 1 până la stâlpul N). Despre fiecare stâlp i se cunoaște $Dist[i]$ ca fiind distanța de la stâlpul curent până la stâlpul 1 - astfel $D[1] = 0$ și $D[N] = L$, iar distanța dintre stâlpul i și stâlpul $i + 1$ este $D[i + 1] - D[i]$.

Ion dorește să strice acest gard (pentru a îl supara pe Vasile) și dorește să elimine cel mult M stâlpi din acesta (dacă ar elimina mai mulți Vasile ar deveni suspicios). Totuși, Ion dorește să facă o pagubă cât mai mare și vrea ca după eliminarea stâlpilor distanța minimă dintre oricare doi stâlpi consecutivi rămași să fie maximă. Cu alte cuvinte, dacă în urma eliminării rămân N' stâlpi ($N' \geq N - M$) se dorește ca $\min_{i=1, N'-1} (D[i + 1] - D[i])$ să fie cât mai mare. **Atenție!**

Ion nu are voie să elimine stâlpul 1 sau stâlpul N .

De asemenea, Ion cunoaște că $M + 2 \leq N \leq 100.000$ și $L \leq 10^9$.

Exemplu: Să presupunem ca avem $N = 5$, $M = 1$, $L = 10$ și distanțele $Dist = \{0, 3, 7, 8, 10\}$. Soluția optimă ar fi să se elimine stâlpul 4 aflat la distanță 8, obținându-se o distanță minimă egală cu 3 - mai mult de atât nu se poate.

Fișierul de intrare pentru această problema va conține pe prima linie trei numere naturale separate printr-un spațiu, N , M și L , respectând restricțiile de mai sus. Pe următoarele N linii va fi dat vectorul $Dist$, pe ce-a de-a i -a linie aflându-se elementul $Dist[i]$. Pentru exemplul de mai sus fișierul de intrare ar fi:

```
5 1 10
0
3
7
8
10
```

Fișierul de ieșire va conține pe prima linie distanța maximă calculată. Pe a doua linie va fi numărul N' de stâlpi rămași după eliminarea a cel mult M stâlpi. Următoarele N' linii vor conține distanțele stâlpilor rămași, pe cea de-a i -a linie aflându-se distanța stâlpului i față de stâlpul 1.

Fișierul de ieșire pentru exemplul de mai sus ar fi:

```
3
4
0
3
```

7
10

3 A doua problemă - Planificare taskuri - 40 de puncte

Lucrând într-o fabrică, ziua de muncă a lui Georgică este împărțită în N sloturi de timp, numerotate de la 1 la N ($N \leq 1000$). Pentru fiecare slot de timp Georgică poate decide dacă să muncească în acel slot sau să ia o pauză de relaxare. Fiecare slot i are asociat un câștig $C[i]$ pe care Georgică îl poate obține dacă ar munci atunci ($-10^4 \leq C[i] \leq 10^4$); acest câștig se dă în felul următor: dacă Georgică se decide să lucreze într-un interval $[i, j]$ atunci câștigul pe care îl obține este $Castig[i, j] = C[i+1] + C[i+2] + \dots + C[j]$ - cu alte cuvinte primul slot nu intră în calcul, deoarece Georgică îl consuma "încălzindu-se" pentru muncă. Într-o zi normală de muncă, Georgică trebuie să lucreze în **exact** T sloturi de timp ($T \leq 1000$) - el își poate planifica munca cum dorește alegându-și un subset de T sloturi din cele N disponibile. Totuși, Georgică dorește să obțină un profit cât mai mare, iar datoria voastră este să îl ajutați și să i-l calculați. Astfel, trebuie să alegeți un set de intervale **disjuncte** $[a_1, b_1], [a_2, b_2], \dots, [a_k, b_k]$ cu $1 \leq a_1 \leq b_1 < a_2 \leq b_2 < \dots < a_k \leq b_k \leq N$ și $\sum_{i=1}^k (b_i - a_i + 1) = T$ astfel încât

$\sum_{i=1}^k Castig[a_i, b_i]$ să fie maximă.

Exemplu: Să presupunem că avem $N = 5$, $T = 4$ și câștigurile $\{3, 9, 1, 1, 7\}$. Soluția optimă ar fi să selectăm intervalele $[1, 2]$ și $[4, 5]$ cu un câștig total de $9 + 7 = 16$.

Fișierul de intrare pentru această problema va conține pe prima linie două numere naturale N și T respectând restricțiile de mai sus. Următoarele N linii conțin câte un număr întreg fiecare, al i -lea număr reprezentând câștigul pentru slotul i . Pentru exemplul de mai sus fișierul de intrare ar fi:

```
5 4
3
9
1
1
7
```

Fișierul de ieșire conține pe prima linie câștigul maxim pe care îl poate obține Georgică. Linia a doua va conține K , numărul de intervale selectate. Următoarele K linii vor conține fiecare un interval $[a_i, b_i]$ ($a_i \leq b_i$) de sloturi de timp în care Georgică va trebui să lucreze astfel încât să obțină câștigul menționat mai sus. Pentru exemplul de mai sus fișierul de ieșire ar fi:

```
16
2
```

1 2
4 5

4 Bonus - 20 de puncte

Problema de bonus reprezintă o generalizare a problemei 2. Cerința rămâne aceeași - singura diferență vine din faptul că se permite ca șirul de sloturi să fie circular. Cu alte cuvinte putem alege și un interval de forma $[a_i, b_i]$ cu $1 \leq b_i < a_i \leq N$, însemnând că vom alege elementele $a_i, a_i + 1, \dots, N, 1, \dots, b_i$ cu câștigul $C[a_i + 1] + \dots + C[N] + C[1] + \dots + C[b_i]$. **Atenție!** Și în acest caz toate intervalele trebuie să fie disjuncte, deci un slot i **NU** poate fi ocupat de mai mult de un interval.

Pentru exemplul de la problema 2, dacă permitem alegerea unui șir circular atunci optim va fi să alegem intervalul $[4, 2]$ cu câștig total $7 + 3 + 9 = 19$. Formatul fișierului de intrare este identic cu cel de la problema doi. Formatul fișierului de ieșire este identic cu cel de la problema 2, cu singura mențiune că putem avea un interval de forma $[a_i, b_i]$ cu $a_i > b_i$, caz care semnifică alegerea unui interval circular.

Pentru inputul de mai sus, fișierul de ieșire ar fi:

19
1
4 2

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. Din cele 5 puncte ale unui test, 3 puncte se vor acorda pentru aflarea valorii corecte (prima linie din fișierul de ieșire) și 2 puncte pentru reconstituirea soluției - punctele pentru reconstituirea soluției sunt condiționate de aflarea valorii corecte.

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:

- 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-pb, 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 1.5 secunde

Limitele de timp pentru problema 2 și problema bonus sunt:

- C/C++ 0.5 secunde
- Java 1.5 secunde