Proiectarea Algoritmilor, Tema 2

Deadline Hard: **25.05.2025 23:55**

Cuprins

0.1 Punctare

- Punctajul total al temei este de 100 de puncte (1p din nota finală). La această temă nu există puncte bonus.
- Tema este compusă din trei probleme. Primele două probleme valorează fiecare 30 de puncte. Ultima problemă valorează 40 de puncte.
- La corectarea manuală se pot aplica depunctări pentru calitatea comentariilor din sursă/README/coding style sau alte aspecte. Pentru exemple de depunctări puteți să vă uitați și peste regulile generale de trimitere a temelor.
- O rezolvare care nu compilează sau nu trece niciun test va fi punctată cu 0 puncte.
- Fiecare problemă are o limită de timp pe test. 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 utile de menționat.
- Checkerul verifică doar existența unui README cu denumire corectă (`README`) și conținut nenul. **Punctajul final pe README și comentarii** se acordă la corectarea manuală a temei.
- Temele pot fi testate automat pe VMchecker Next. Acesta suportă temele rezolvate în C/C++ și Java.
- Arhiva cu rezolvarea temei trebuie să fie `.zip`, și va conține:
  - Fișierul/fișierele sursă
  - Fișierul `README`
- **Punctajul final pe teste** este cel obținut în urma rulării pe VMchecker Next.

  ATENȚIE! Tema va fi compilată și testată **DOAR pe Linux**.

  ATENȚIE! Orice nerespectare a restricțiilor duce la pierderea punctajului (după regulile de mai sus).

0.1.1 Links

- Debugging și Structuri de Date
- Coding Tips pentru teme la PA (OCW)

1 Problema 1

1.1 Enunț

Fie un vector cu N numere întregi $v_1, v_2, \ldots, v_N$. Se cere construirea unui graf neorientat cu N noduri pentru care, daca pornim o parcurgere în lățime din nodul 1, vectorul de distanțe obținut coincide cu v.
ATENȚIE! Vor fi acceptate doar soluțiile care folosesc cel mult 10^6 muchii.

1.2 Date de intrare

Pe prima linie a fișierului **p1.in** se află un număr întreg, N.
Pe următoarea linie se află N numere întregi $v_1, v_2, v_3, \ldots, v_N$ reprezentând vectorul de distanțe dorit.

1.3 Date de ieșire

Fișierul **p1.out** va conține pe prima linie o valoare întreagă M, reprezentând numărul de muchii din graf. Următoarele M linii vor avea câte două numere întregi, separate printr-un singur spațiu, care reprezintă capetele muchiilor.

1.4 Restricții și precizări

- $1 \leq N \leq 10^5$
- $0 \leq v_1, v_2, \ldots, v_N \leq N$
- E garantat că $v_1 = 0$
- Nodurile sunt numerotate cu numere întregi între 1 și N.
- Dacă nu există un graf pentru care să se obțină vectorul respectiv de distanțe, se va afișa -1.

1.5 Testare și punctare

- Sursa care conține funcția **main** trebuie obligatoriu denumită: **p1.c**, **p1.cpp** sau **p1.java**.

1.6 Exemple

p1.in	p1.out	Explicație
4 0 1 1 2	3 1 2 1 3 3 4	Pentru a ajunge din nodul 1 în nodul 2 avem muchia directă 1-2. Pentru a ajunge din nodul 1 în nodul 3 avem muchia directă 1-3. Pentru a ajunge din nodul 1 în nodul 4 avem lanțul 1-3-4.

2 Problema 2

2.1 Enunț

Se dă o matrice de dimensiuni $N \times M$ cu valori întregi. Se cere aria celei mai mari zone de casuțe conectate pentru care diferența dintre valoarea maximă și valoarea minimă din acea zonă este de cel mult K.

2.2 Date de intrare

Pe prima linie a fisierului *p2.in* se află trei numere intregi, N, M, respectiv K.
Pe următoarele N linii se vor afla câte M numere separate prin spații, unde al j-lea număr de pe a i-a linie reprezintă $c_{i,j}$, valoarea aflată pe pozitia (i, j) în matrice.

2.3 Date de ieșire

În fișierul *p2.out* se va scrie aria maximă a zonei găsite.

2.4 Restricții și precizări

- $1 \leq N, M \leq 100$
- $0 \leq K \leq 10^9$
- $0 \leq c_{i,j} \leq 10^9$
- Definim aria unei zone de căsuțe conectate ca fiind numărul de căsuțe care fac parte din zona respectivă.
- Două casuțe sunt conectate dacă sunt vecine pe verticală sau pe orizontală.

2.5 Testare și punctare

- Sursa care conține funcția **main** trebuie obligatoriu denumită: **p2.c**, **p2.cpp** sau **p2.java**.

2.6 Exemple

p2.in	p2.out	Explicație
3 3 2 1 2 4 1 1 5 0 10 5	5	Valorile din zonele pe care le putem găsi sunt: 1. 0, 1, 1, 1, 2 2. 2, 4 3. 4, 5, 5 4. 10 Se observă că cea mai mare zonă conține 5 casuțe.

3 Problema 3

3.1 Enunț

Pe drumul spre un jaf de zile mari, Robin Hood o aude pe dragostea vieții sale, Maid Marian, tipand din mijlocul lacului. Neavând o barca, dar fiind ingenios, el vrea să ajungă până la Maid Marian sărind pe bușteni care plutesc pe lac. Totuși, din cauza curentilor, bușteni își schimbă poziția încontinuu, așa că Robin Hood are nevoie de ajutorul vostru pentru a își calcula traseul.

Robin Hood știe că are la dispoziție timpul T pentru a ajunge la domniță, înainte ca aceasta să se înece. De asemenea, el știe că pe lac sunt N bușteni, iar pentru fiecare dintre acestia el poate vedea pozitiile initiale ale capetelor: $(x_i^{start}, y_i^{start}); (x_i^{end}, y_i^{end}), i \in \{1, \ldots, N\}$.

În plus, cunoscând acel lac ca pe propriul lui buzunar, el știe deja pentru fiecare buștean în ce direcție se va misca acesta la fiecare moment de timp. $m_i^j, i \in \{1, \ldots, N\}, j \in \{0, \ldots T-1\}$, reprezinta direcția în care se va mișca bușteanul i la momentul de timp j și poate fi una din direcțiile N, S, E, V. Se consideră că un buștean se mișca la un moment dat cu exact o unitate.

Robin Hood începe de pe primul capăt al busteanului 1 la timpul 0. La fiecare moment de timp, Robin Hood are o decizie de luat:

1. El poate sta pe loc, relativ la bușteanul pe care se află. In acest caz, va consuma E_1 energie și se va mișca odata cu busteanul.
2. El poate să facă un pas pe buștean. În acest caz va consuma E_2 energie și se va deplasa cu o unitate în una din direcțiile N, S, E, V. Pentru un buștean aflat în pozitie verticală doar mișcarile N și S sunt disponibile, iar pentru un buștean aflat in pozitie orizontala, doar E și V. Dacă Robin se află într-unul din capetele bușteanului, una din cele 2 mutări nu va fi disponibilă, întrucât Robin ar cădea în apă.
3. Uneori, doi bușteni se pot intersecta (din cauza curenților, un buștean poate ajunge să treacă pe sub altul). În aceasta situație, Robin are și opțiunea de a sări de pe un buștean pe celălalt. Cu alte cuvinte, dacă la momentul t Robin se află pe un buștean B_1 în punctul de intersecție cu un alt bustean B_2, Robin poate alege să sară pe B_2, tot in punctul de intersecție cu B_1, urmând să se miște în continuare pe lac odata cu B_2. În acest caz, energia consumată de Robin va fi E_3.

Întrucât nu contează cât timp îi ia lui Robin să ajungă la Maid Marian, atât timp cât se încadrează în timpul T, el își dorește să o salveze consumând cât mai puțină energie, pentru a putea apoi să continue și cu jaful.

Nefiind foarte priceput la calcule, el vă cere vouă să îi indicați energia minimă pe care trebuie să o consume pentru a o salva pe Maid Marian, precum și mișcarile pe care ar trebui să le facă.

3.2 Date de intrare

Pe prima linie a fisierului **p3.in** se găsesc 2 întregi T și N, reprezentând timpul pe care il are Robin la dispoziție, respectiv numărul de bușteni de pe lac.

Pe cea de-a doua linie se găsesc 2 întregi reprezentând coordonatele la care se află Maid Marian.

Pe cea de-a treia linie se găsesc 3 întregi reprezentând energiile E_1, E_2 și E_3 corespunzătoare celor trei tipuri de acțiuni posibile.

Pe următoarele N linii se găsesc câte 4 întregi, reprezentând coordonatele capetelor bușteni lor: x_i^{start}, y_i^{start}, x_i^{end}, y_i^{end}, $i \in \{1 \ldots N\}$.

Pe următoarele N linii se găsesc câte T caractere, reprezentând direcția în care se vor mișca bușteni i la fiecare moment de timp (m_i^j, $i \in \{1, \ldots, N\}$, $j \in \{0, \ldots T-1\}$).

3.3 Date de ieșire

Pe prima linie a fișierului **p3.out** se va afișa un întreg reprezentând energia minimă pe care trebuie să o consume Robin Hood pentru a reuși să o salveze pe Maid Marian la timp.

Pe a doua linie se va afla numărul de mișcări M pe care trebuie să le facă Robin pentru a ajunge la destinație. Vor urma apoi M linii care vor conține cele M mișcări:

1. Pentru o acțiune de tipul "stat pe loc" se va afișa caracterul H.
2. Pentru o acțiune de mișcare pe bușteanul curent în una din direcțiile cardinale se va afișa caracterul corespunzător direcției respective: $N/S/E/V$
3. Pentru o acțiune de tipul "sărit pe alt buștean" se va afișa caracterul J urmat de un spațiu liber și apoi de indexul bușteanului pe care trebuie să sară Robin.

3.4 Restricții și precizări

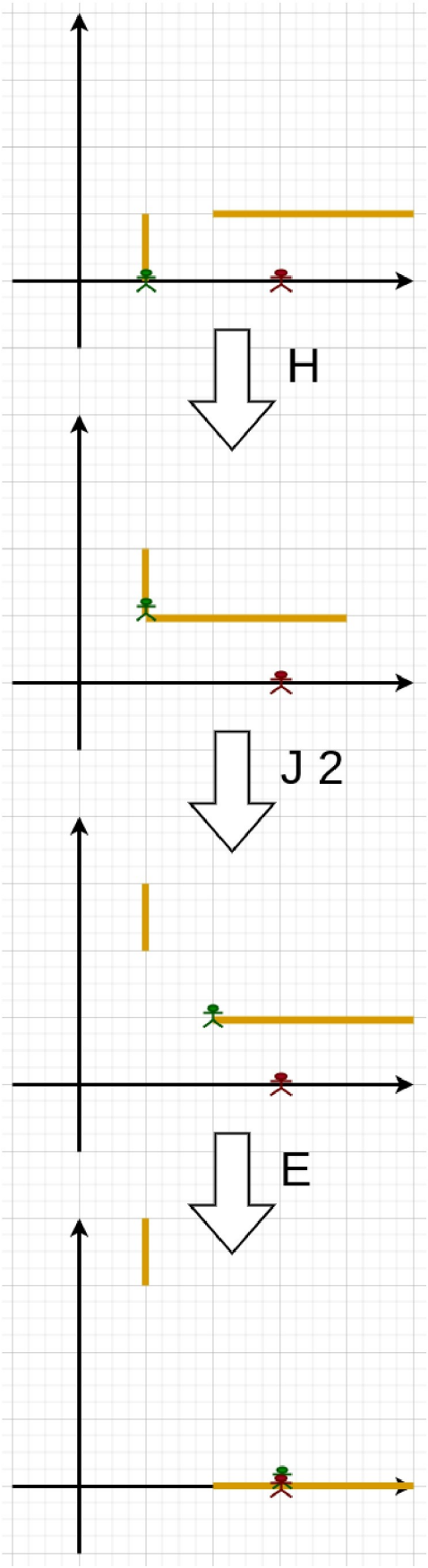
- $1 \leq N \leq 80$
- $1 \leq T \leq 400$
- $1 \leq L \leq 10$
- $1 \leq E_1, E_2, E_3 \leq 1000$
- $-500 \leq x_i^{start}, x_i^{end}, y_i^{start}, y_i^{end} \leq 500$
- Bușteanii vor avea întotdeauna orientarea verticală sau orizontală ($x_i^{start} = x_i^{end}$ sau $y_i^{start} = y_i^{end}, \forall i \in \{1, \ldots, N\}$)
- $x_i^{start} \leq x_i^{end}$ și $y_i^{start} \leq y_i^{end}$
- $m_i^j \in \{N, S, E, V\}$; $\forall i \in \{1, \ldots, N\}, \forall j \in \{0, \ldots, T-1\}$
- Indexarea bușteni lor se va considera de la 1 la N.
- Coordonatele se consideră în sens matematic: prin mișcare la N se înțelege creșterea coordonatei y, iar prin mișcare la E creșterea coordonatei x.
- Orice soluție care obține efortul minim va fi acceptată.

3.5 Testare și punctare

Sursa care conține funcția **main** trebuie obligatoriu denumită: **p3.c**, **p3.cpp** sau **p3.java**.

3.6 Exemple

p3.in	p3.out	Explicație
3 2 3 0 1 2 3 1 0 1 1 2 1 5 1 NNN VES	6 3 H J 2 E	Urmăriți imaginea de pe pagina următoare. Robin Hood este reprezentat folosind culoarea verde, iar Maid Mirian folosind culoarea roșie.


H
J 2
E