

Structuri de date - Seria CC

Livrare de produse folosind drone

Tema 3

Deadline hard: 14 Mai 2017

Iulian-Gabriel Radu

Adrian Salomia

24 aprilie 2017

1 Introducere

În prezent, livrarea produselor cu drone devine din ce în ce mai accesibilă. Dexter este un producător de produse bio. El dorește să își extindă afacerea prin livrarea produselor către clienți cu ajutorul unor drone. Fiind la început de drum, el deține o singură dronă și te-a rugat pe tine să îl ajuți cu un algoritm pe care drona îl poate folosi pentru a livra produsele într-un mod cât mai eficient.

2 Descrierea problemei

Spațiul aerian este compus din culoare aeriene unidirecționale între puncte de interes, ce au o distanță cunoscută.

Drona folosită de Dexter are capacitate de livrare limitată. În orice moment ea poate avea produse doar pentru un singur client.

Fermele sunt punctele de ridicare ale produselor. Ele pot fi de mai multe tipuri și au stoc nelimitat.

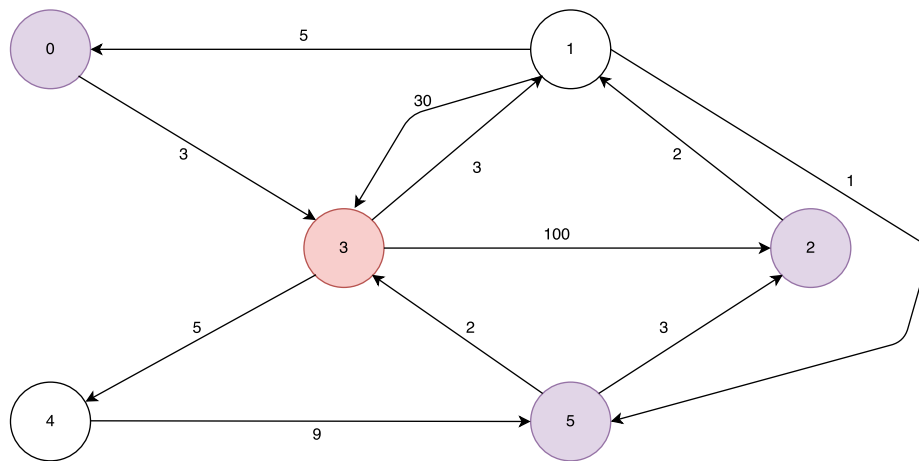
Clienții potențiali reprezintă puncte de interes ce se găsesc la capetele culoarelor aeriene.

O **comandă** reprezintă tipul de produs ce trebuie livrat unui anumit client.

3 Task-uri

3.1 Task 1

Pentru acest task, presupunem că există o singură fermă. Să se proiecteze un algoritm ce permite dronei să livreze tuturor clienților produsele pe cel mai scurt drum posibil. Comenzile trebuie procesate în ordinea în care au fost efectuate de către clienți. Punctul de plecare al dronei este ferma. Din cauza faptului că drona este limitată la a deține produse pentru un singur client, ea trebuie să se întoarcă la fermă pentru aprovizionare după fiecare comandă procesată. După procesarea tuturor comenzilor, drona trebuie să se întoarcă înapoi în punctul de plecare.



Input:

```
1
6
0 0 0 1 0 0
11
0 3 3
1 0 5
1 3 30
1 5 1
2 1 2
3 1 3
3 2 100
3 4 5
4 5 9
5 2 3
5 3 2
3
1 2
1 5
```

1 0

Output:

3 1 5 2 1 5 3 1 5 3 1 0 3
29

Format input:

- Pe prima linie se găsește un întreg, ID, ce reprezintă un identificator pentru task-ul ce trebuie rezolvat.
- Pe a 2-a linie se găsește un întreg, N, ce reprezintă numărul de puncte de interes.
- Pe a 3-a linie se găsește o listă de N întregi, ce reprezintă tipul fiecărui punct de interes (potențial client - 0, fermă - 1).
- Pe a 4-a linie se găsește un întreg, M, ce reprezintă numărul de culoare aeriene disponibile.
- Pe următoarele M linii, se găsesc câte 3 numere. Primele două reprezintă capetele culoarelor aeriene, iar al 3-lea număr, lungimea culoarului.
- Pe linia imediat următoare, se găsește un întreg C, ce reprezintă numărul de comenzi.
- Pe următoarele C linii, se găsesc perechi de întregi formate din tipul fermei de la care trebuie ridicate produsele și identificatorul clientului la care trebuie livrate.

Format output:

Fișierul de output conține 2 linii:

- Pe prima linie, separați prin spațiu, identificatorii nodurilor punctelor de interes prin care drona trece pentru procesarea tuturor comenzilor.
- Pe a doua linie, un întreg, ce reprezintă distanța totală parcursă.

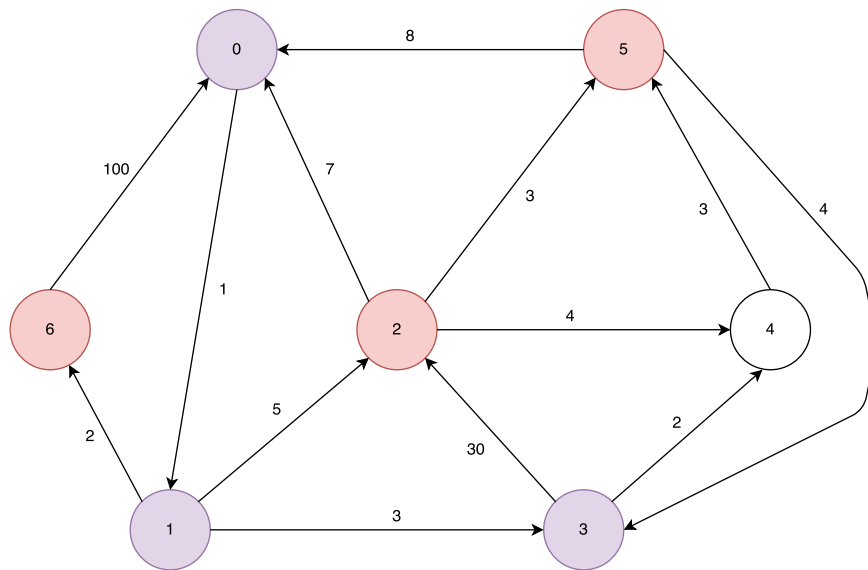
Explicație output:

Punctul de plecare al dronei este ferma. Prima comandă trebuie livrată către clientul 2. Se caută cel mai scurt drum către 2 și se găsește drumul 3, 1, 5, 2 de cost 7. Fiind acum în punctul 2, drona trebuie să se aprovizioneze pentru următoarea comandă, cu produse de tip 1 (o singură fermă de tip 1, cu identificatorul 3) deci merge pe cel mai scurt drum până la fermă - 1, 5, 3 de cost 5. După aprovizionare, în mod similar, procesează și următoarele 2 comenzi, către clientul 5 și către clientul 0. În final, este afișat drumul complet și costul total, 29.

4 Task 2

După o perioadă de succes în livrarea produselor de la o singură fermă, Dexter a decis că are nevoie de un nou algoritm pentru livrarea comenzilor, astfel încât drona să se poată aproviziona de la orice fermă disponibilă.

Astfel, ca și la task-ul 1, există un singur tip de fermă, însă, de această dată, există mai multe ferme de acest tip. De asemenea, drona se poate aproviziona de la orice fermă. În același mod ca și la primul task, drona trebuie să livreze tuturor clienților produsele pe cel mai scurt drum posibil, în ordinea în care comenzile au fost efectuate, iar după finalizarea acestora să se întoarcă în cel mai apropiat depozit.



Input:

```
2
7
0 0 1 0 0 1 1
13
0 1 1
1 2 5
1 3 3
1 6 2
2 0 7
2 4 4
2 5 3
3 2 30
3 4 2
4 5 3
```

```
5 0 8
5 3 4
6 0 100
3
1 3
1 1
1 0
2
```

Output:

```
2 5 3 4 5 0 1 2 0 1 6
36
```

Format input:

- Pe prima linie se găsește un întreg, ID, ce reprezintă un identificator pentru task-ul ce trebuie rezolvat.
- Pe a 2-a linie se găsește un întreg, N, ce reprezintă numărul de puncte de interes.
- Pe a 3-a linie se găsește o listă de N întregi, ce reprezintă tipul fiecărui punct de interes (potențial client - 0, fermă - 1).
- Pe a 4-a linie se găsește un întreg, M, ce reprezintă numărul de culoare aeriene disponibile.
- Pe următoarele M linii, se găsesc câte 3 numere. Primele două reprezintă capetele culoarelor aeriene, iar al 3-lea număr, lungimea culoarului.
- Pe linia imediat următoare, se găsește un întreg C, ce reprezintă numărul de comenzi.
- Pe următoarele C linii, se găsesc perechi de întregi formate din tipul fermei de la care trebuie efectuată comanda și identificatorul clientului la care trebuie livrată.
- *Pe ultima linie, se găsește un întreg ce reprezintă indicele fermei de start.*

Format output:

Fișierul de output conține 2 linii:

- Pe prima linie, separați prin spațiu, identificatorii nodurilor punctelor de interes prin care drona trece pentru procesarea tuturor comenzilor.
- Pe a doua linie, un întreg, ce reprezintă distanța totală parcursă.

Explicație output:

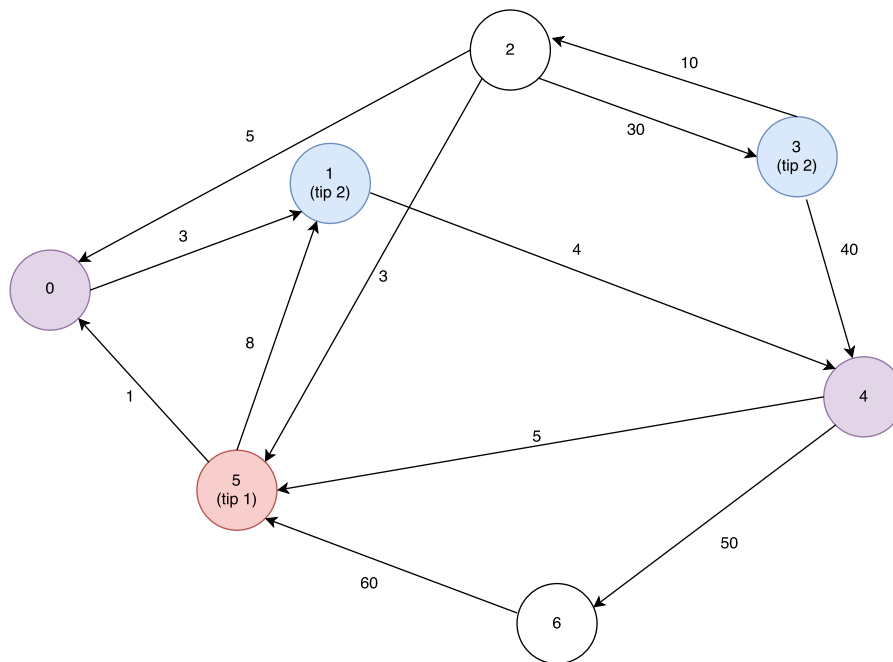
Punctul de plecare al dronei este punctul de interes cu identificatorul 2, citit din fișierul de intrare. Prima comandă trebuie livrată către clientul 3 și este realizată pe cel mai scurt

drum posibil. După procesarea acestei comenzi, drona trebuie să ajungă la următorul client, cu identificatorul 1, pe drumul scurt, trecând printr-o fermă pentru aprovizionare. În mod similar, se procesează și restul comenzilor, se afișează calea completă și costul total al drumurilor. După efectuarea ultimei comenzi, către nodul 0, drona se întoarce la ferma cea mai apropiată, cea cu identificatorul 6.

5 Task 3

Se dorește din nou îmbunătățirea algoritmului astfel încât drona să poată livra produse de la orice tip de fermă, în funcție de comenzile clienților.

Pentru acest task, ca și la task-ul 2, există mai multe ferme, însă, de această dată, fermele pot fi de tipuri diferite. Pentru fiecare comandă este specificat tipul fermei la care se găsește produsul. Comenzile nu mai trebuiesc procesate în ordinea în care au fost efectuate, ci într-o ordine ce garantează că produsele vor fi livrate pe cel mai scurt drum posibil. Ca și la task-ul anterior, drona trebuie să livreze tuturor clienților produsele pe cel mai scurt drum posibil, iar după finalizarea acestora, să se întoarcă la cea mai apropiată fermă.



Input:

```

3
7
0 2 0 2 0 1 0
12

```

```
0 1 3
1 4 4
2 0 5
2 3 30
2 5 3
3 2 10
3 4 40
4 5 5
4 6 50
5 0 1
5 1 8
6 5 60
2
1 0
2 4
3
```

Output:

```
3 2 5 0 1 4 5
26
```

Format input:

- Pe prima linie se găsește un întreg, ID, ce reprezintă un identificator pentru task-ul ce trebuie rezolvat.
- Pe a 2-a linie se găsește un întreg, N, ce reprezintă numărul de puncte de interes.
- Pe a 3-a linie se găsește o listă de N întregi, ce reprezintă tipul fiecărui punct de interes (potențial client - 0, fermă de primul tip - 1, fermă de al doilea tip - 2, etc.).
- Pe a 4-a linie se găsește un întreg, M, ce reprezintă numărul de culoare aeriene disponibile.
- Pe următoarele M linii, se găsesc câte 3 numere. Primele două reprezintă capetele culoarelor aeriene, iar al 3-lea număr, lungimea culoarului.
- Pe linia imediat următoare, se găsește un întreg C, ce reprezintă numărul de comenzi.
- Pe următoarele C linii, se găsesc perechi de întregi formate din tipul fermei de la care trebuie efectuată comanda și identificatorul clientului la care trebuie livrată.
- *Pe ultima linie, se găsește un întreg ce reprezintă indicele fermei de start.*

Format output:

Fișierul de output conține 2 linii:

- Pe prima linie, separați prin spațiu, identificatorii nodurilor punctelor de interes prin care drona trece pentru procesarea tuturor comenzilor.
- Pe a doua linie, un întreg, ce reprezintă distanța totală parcursă.

Explicație output:

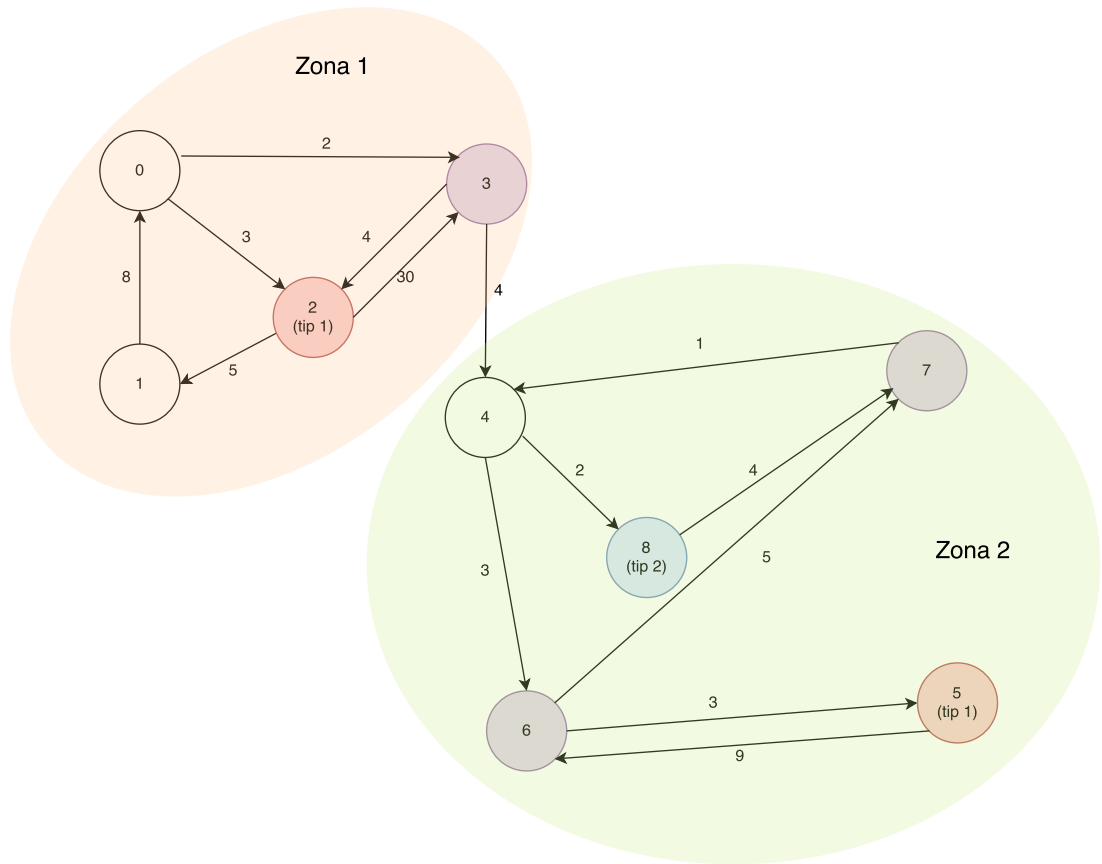
Punctul de plecare al dronei este punctul de interes cu identificatorul 3, citit din fișierul de intrare. De această dată, comenzile nu mai trebuie procesate în ordinea în care au fost efectuate. De aceea, se caută ordinea de procesare ce garantează că drumul parcurs de dronă pentru efectuarea tuturor comenzilor va fi drumul cel mai scurt.

6 Bonus

Pentru acest task, există mai multe zone de livrare și un număr de drone egal cu numărul acestor zone. O zonă de livrare are proprietatea că între oricare două puncte de interes, există un culoar aerian. Pentru fiecare astfel de zonă, se va plasa câte o dronă pe ferma cu identificatorul cel mai mic, și se va calcula un număr ce reprezintă profitabilitatea zonei de livrare. Se vor afișa, pentru fiecare zonă, nodurile ce fac parte din aceasta și profitabilitatea.

Ca și la task-ul anterior, există mai multe ferme și mai multe tipuri de ferme, tipul fermei de la care trebuie ridicat un produs se găsește în comandă, iar drona trebuie să livreze tuturor clienților produsele pe cel mai scurt drum posibil. După finalizarea procesării comenzilor, drona trebuie să se întoarcă la cea mai apropiată fermă de destinația finală. Drona nu va părăsi niciodată zona în care a fost plasată.

Formula pentru calcularea profitabilității unei zone este $\text{alfa} * \text{numar_clieni} - \text{beta} * \text{distanța}$, unde alfa și beta sunt citite din fișierul de input.



Input:

```

4
9
0 0 1 0 0 1 0 0 2
14
0 2 3
0 3 2
1 0 8
2 1 5
2 3 30
3 2 4
3 4 4
4 6 3
4 8 2
5 6 9
6 5 3

```

```
6 7 5
7 4 1
8 7 4
3
1 3
1 7
2 6
20 1
```

Output:

```
2
4 5 6 7 8 12
0 1 2 3 1
```

Format input:

- Pe prima linie se găsește un întreg, ID, ce reprezintă un identificator pentru task-ul ce trebuie rezolvat.
- Pe a 2-a linie se găsește un întreg, N, ce reprezintă numărul de puncte de interes.
- Pe a 3-a linie se găsește o listă de N întregi, ce reprezintă tipul fiecărui punct de interes (potențial client - 0, fermă de primul tip - 1, fermă de al doilea tip - 2, etc.).
- Pe a 4-a linie se găsește un întreg, M, ce reprezintă numărul de culoare aeriene disponibile.
- Pe următoarele M linii, se găsesc câte 3 numere. Primele două reprezintă capetele culoarelor aeriene, iar al 3-lea număr, lungimea culoarului.
- Pe linia imediat următoare, se găsește un întreg C, ce reprezintă numărul de comenzi.
- Pe următoarele C linii, se găsesc perechi de întregi formate din tipul fermei de la care trebuie efectuată comanda și identificatorul clientului la care trebuie livrată.
- Pe ultima linie, 2 întregi, ce reprezintă coeficienții alfa și beta din formulă.

Format output:

- Pe prima linie, un întreg, Z, ce reprezintă numărul de zone
- Pe următoarele Z linii, câte o listă de întregi, separați prin spațiu, sortați în ordine crescătoare ce reprezintă punctele de interes ce fac parte dintr-o anumită zonă și un întreg ce reprezintă profitabilitatea zonei. Cele Z linii, vor fi ordonate în ordine descrescătoare în funcție de profitabilitate.

Explicație output:

1.
 - prima zonă este formată din următoarele puncte: 0, 1, 2, 3
 - drona pleacă din punctul de interes cu identificatorul 2
 - traseul dronei este următorul: 2 1 0 3 2 (distanța = $5 + 8 + 2 + 4 = 19$)
 - profitabilitatea este: $20 \cdot 1 - 1 \cdot 19 = 1$
2.
 - a doua zonă este formată din următoarele puncte: 4, 5, 6, 7, 8
 - drona pleacă din punctul de interes cu identificatorul 5
 - traseul dronei este următorul: 5 6 7 4 8 7 4 6 5 (distanța = $9 + 5 + 1 + 2 + 4 + 1 + 3 + 3 = 28$)
 - profitabilitatea este: $20 \cdot 2 - 1 \cdot 28 = 12$

7 Restricții și precizări

Temele trebuie încărcate pe site-ul vmchecker. NU se acceptă teme trimise prin alte mijloace decât prin intermediul vmchecker.

O rezolvare constă dintr-o arhivă de tip **zip** care conține toate fișierele sursă, un fișier **Makefile**, cu regulile **build** și **clean** și un **README** în care se vor preciza detaliile de implementare. Regula **build** trebuie să aibă ca efect compilarea surselor și crearea binarului **livrare**.

Executabilul va primi ca parametri în linia de comandă numele de fișierului de intrare și numele fișierului de ieșire:

`./livrare [fișier_intrare] [fișier_ieșire]`

8 Punctaj

Task	Punctaj
Task 1	35 puncte
Task 2	30 puncte
Task 3	25 puncte
Bonus	20 puncte
Coding Style, README, warning-uri	10 puncte