

## Tema 2 – SDA

### Marele arhipelag Eao-Zinci

#### Descriere

Arhipelagul Eao-Zinci din Orientul Indepartat este in plina dezvoltare. Fiecare insula din arhipelag produce zilnic o serie de resurse. Acestea sunt folosite local si/sau distribuite catre alte insule folosind flota aeriana Eao-Zinci.

Rutele de transport sunt configurate in fiecare dimineata pentru a asigura distribuția optimă a resurselor (etapa de proiectare).

Intrand in era automatizării, conducatorii arhipelagului doresc sa treacă la următorul pas si să poata sa-si distribuie resursele in cel mai eficient mod posibil (etapa de analiza).

Mai mult, pentru a elimina potențialele conflicte in ce privește accesul la flota aeriana, in fiecare seara, avioanele sunt redistribuite între insule (etapa de gestiune).

Pentru a rezolva aceste probleme, conducatorii cauta un programator priceput si au formulat problemele lor mai jos!

#### Input

Veti avea la dispozitie un fisier de tipul **date.in** care va avea urmatoarea structura:

- 1) Numar de insule - n
- 2) Pe urmatoarele n linii se vor descrie insulele cu resursele proprii.

Nume insula

Numar resurse –m

Tip\_1 resursa si cantitate

Tip\_2 resursa si cantitate

...

Tip\_m resursa si cantitate

Exemplu:

Island1

3

Fier 350

Carbune 150

Piatra 100

Island2

3

Argint 200

Aur 10

Fier 350

- 3) Numar conexiuni – p (se vor prezenta conexiunile intre insulele arhipelagului si durate de zbor prestabilite).

- 4) Pe urmatoarele p linii se vor afisa conexiunile intre insule: IslandY - IslandZ durata\_zbor

Exemplu:

Island1 - Island3 30

Island1 - Island5 25

Observatie: Daca exista conexiune intre insula X si insula Y, atunci exista un zbor care duce din X in Y, dar si un zbor care duce din Y in X

5) **Cerinte partea de proiectare si analiza** (se citeste fluxul de date pana la terminarea fisierului in cazul testelor 1-8 si, respectiv, pana la intalnirea primei linii care contine doar o valoare intreaga in cazul testelor 9-14). Cerintele pentru etapele de proiectare si analiza pot sa fie:

- \* conexiune X Y
- \* adauga\_zbor X Y cost
- \* anulare\_zbor X Y
- \* legatura X
- \* alte\_cerinte (vor fi prezentate mai jos)

Exemplu:

conexiune Island1 Island3

adauga\_zbor Island2 Island3 15

anulare\_zbor Island3 Island5

## **Descriere cerinte**

Dupa citirea insulelor si conexiunilor, fisierul de input va contine partea de proiectare (modificarea traseelor intre insule pentru ziua curenta) si cea de analiza a datelor pentru distribuirea optima a resurselor. Astfel:

### 1. Proiectare

Avand in vedere dezvoltarea continua a arhipelagului in fiecare zi si intentia acestora de automatizare a distribuirii resurselor catre celalalte insule in nevoie, se comunica niste modificari generale.(obs: intre parantezele drepte se afla formatul pe care o sa-l gasiti in fisierul date.in)

- **[conexiune IslandX IslandY]** : Se va afisa **OK** daca exista conexiune directa intre cele doua insule si **NO** in caz contrar
- **[legatura IslandX]** : se cere **legatura directa** a Insulei X cu celelate insule din arhipelag  
**OBS: In cazul in care nu exista nicio legatura NU se afiseaza nimic.**
- **[adauga\_zbor IslandX IslandY cost]** : se va adauga un **drum direct** de durata **cost** intre Insula X si Insula Y  
Observatie: **cost** reprezinta durata in minute.
- **[anulare\_zbor IslandX IslandY]**: se va sterge drumul direct intre Insula X si Insula Y
- **[max\_resurse]**: se va afisa numarul de resurse distincte de pe tot arhipelagul, urmat de numele resurselor separate printr-un spatiu in ordine lexicografica

**Exemplu(pe structura de mai sus): 5 Argint Aur Carbune Fier Piatra**

- **[max\_cantitate resursa]**: se va afisa insula (respectiv insulele) cu cantitatea cea mai mare de resursa disponibila (OBS: Afisarea insulelor se va face in ordine -de la prima la ultima-)

**Exemplu(pe structura de mai sus): Pentru max\_cantitate Fier se va afisa Island1 Island2**

### 2. Analiza

In urmatoarea parte, se vor dezvolta algoritmi pentru determinarea eficienta a drumurilor, respectiv timpului intre doua insule.

- **[drum\_zbor IslandX IslandY]** : se cere afisarea drumului cel mai scurt (din punct de vedere al duratei) de la Insula X la Insula Y. Afisarea va fi de la prima insula intermediara pana la insula destinatie, adica Insula Y.(Obs: parcurgerea insulelor va fi separata printr-un spatiu).

**OBS: In cazul in care NU exista drum se va afisa NO**

**In contextul minimizarii timpului de zbor intre doua insule, se vor analiza doua scenarii:**

- **[timp\_zbor IslandX IslandY]**: se cere afisarea duratei minime in conditii de distribuire instantanea

**OBS: In cazul in care NU se poate determina timpul zborului se va afisa INF.**

*Acesta este cazul in care se distribuie resursele in timpul diminetii, cand traficul nu este mare, considerandu-se schimbul de resurse instant(adica fara timp de asteptare).*

- **[min\_zbor IslandX IslandY]:** se cere afisarea duratei minime in conditiile adaugarii stationarii pe insulele intermediare pana la Insula Y

*Acesta este cazul in care se distribuie resursele atunci cand traficul este ingreunat, considerandu-se o stationare standard de 15 minute pe fiecare insula, pentru atenuarea coliziunilor sau supraincarii aeroporturilor insulelor.*

**OBS1:** Stationarea este aceeași la toate insulele și se adaugă o dată ce avionul aterizează pe insula respectivă.

**OBS2:** Acest caz reprezintă un test separat, Testul 8, considerandu-se in restul testelor cazul favorabil, atunci cand distributia resurselor se poate face instant.

**Exemplu :**

Presupunand ca avem urmatoarele date initiale :

Island1 - Island3 40

Island1 - Island2 10

Island2 - Island3 25

Pentru **drum\_zbor Island1 Island3** se va afisa **Island2 Island3**

Pentru **timp\_zbor Island1 Island3** se va afisa **35** (10+25)

Pentru **min\_zbor Island1 island3** se va afisa **40** (deoarece daca am fi ales drumul prin Island2 s-ar fi adaugat timpul de stationare de 15 minute si ar fi devenit 50 > 40)

### 3. Gestiune

Pentru buna dispozitie a locuitorilor arhipelagului si evitarea razboaielor intre insule pe tema accesului facil la resurse si flota aeriana, conducatorii au hotarat urmatoarele: :”deoarece la finalul zilei pot sa existe insule cu un aerodrom sub sau supraincarcat, flota se redistribuie in fiecare seara, daca e cazul.”

Pentru optimizarea problemei, conducatorul arhipelagului, Ghi Yu, cere o analiza a tuturor posibilitatilor de gestionare a avioanelor, introducand atat restrictii, cat si evenimente recurente. In urma analizei, se doreste diminuarea sub/supraincarii: fiecare insula poata sa aiba un numar **maxim** de avioane, **toleranta** maxima e comunicata tuturor si aceeași pentru toate insulele.

Testele 9 - 14 contin, dupa cerintele legate de proiectare si analiza, datele de intrare si cerintele pentru partea de proiectare organizate astfel:

Numar de insule n

Toleranta t

Numar de avioane pentru fiecare insula (vector de dimensiune n)

Pe fiecare dintre urmatoarele n linii se da vectorul cu id-urile avioanelor insulei X (id-urile sunt ordonate crescator)

Pe urmatoarele n linii se gaseste matricea de adiacenta care precizeaza existenta conexiunilor directe intre insule ( 1 - exista conexiune si cu 0 - nu exista conexiune intre insule)

### Exemplu

5  
4  
1 6 2 1 3  
1  
3 9 17 18 22 98  
4 7  
12  
19 21 23  
0 1 0 0 1  
1 0 1 1 0  
0 1 0 1 0  
0 1 1 0 0  
1 0 0 0 0

**OBS:** Etapa de gestiune (backtracking) este tratata **individual** (legaturile intre insule sau numarul insulelor poate sa fie diferit fata de prima parte a fisierului cu cerinte - etapele de analiza si proiectare).

Dupa prezentarea situatiei curente pentru fiecare insula, se prezinta urmatoarele cazuri posibile care pot aparea in contextul gestionarii avioanelor.

#### *Cazuri posibile de output:*

- 1) Nu trebuie distribuit niciun avion => Se va afisa input-ului de la cerinta curenta (nr insule, toleranta, etc - exact in forma gasita in fisierul date.in)
- 2) Exista un numar prea mare de avioane, iar problema nu poate fi rezolvata => Se va afisa mesajului: „Stack overflow!”
- 3) O insula distribuie avioane celor cu care se afla in legatura directa => Se va afisa fiecare aranjament posibil pentru fiecare insula
- 4) Mai multe insule distribuie avioane, dar nu intra in conflict => Similar punctul anterior
- 5) Mai multe insule distribuie avioane, dar intra in conflict => prioritatea este data de ordinea insulelor, in mod crescator
- 6) O insula distribuie avioane, dar insulele cu care este conectata nu au capacitatea necesara sa accepte atatea avioane => in ordine crescatoare, in functie de legaturi, prima insula o sa distribuie avioane pentru a facilita locurile necesare

**OBS:** Insula secundara care trebuie acum sa faca distribuirea o sa faca un singur drum (nu se cer aranjamentele).

#### Exemplu:

InsulaX are avioanele 3, 5, 7, 8, 22, 33, iar toleranta este 4 => trebuie sa distribuie avioanele cu id-ul 22 si 33.

Aceasta are legatura cu urmatoarele insule:

InsulaB cu avioanele 40, 41, 42, 43

InsulaC cu avioanele 50, 51, 52, 53

Se poate observa ca acestea si-au atins toleranta. Pentru a indeplini cerinta de distribuire, se verifica, incepand de la prima insula cu care are legatura, daca aceasta poate sa distribuie numarul necesar de avioane pentru a facilita distribuirea.

InsulaB are conexiuni cu insulaD si insulaE

insulaD cu avioanele 90, 91, 92, 93

InsulaE cu avioanele 80, 81, 82, 83

Deoarece nu exista cel putin 2 locuri libere (numarul de avioane cedate de insulaX), trecem mai departe.

InsulaC are conexiuni cu insulaF si insulaG:

insulaF cu avioanele 1, 2  
 insulaG cu avioanele 61, 62  
 Deoarece insulaF are 2 locuri disponibile, transferul o sa fie urmatorul:  
 InsulaC cedeaza 2 avioane insulei F:  
 InsulaF are avioanele 1, 2, 52, 53  
 InsulaX poate sa isi cedeze avioanele insuleiC:  
 InsulaC are avioanele 22, 33, 50, 51  
 Deci, output-ul o sa arate astfel:  
 InsulaB  
 40, 41, 42, 43  
 InsulaC  
 22, 33, 50, 51  
 InsulaD  
 90, 91, 92, 93  
 InsulaE  
 80, 81, 82, 83  
 InsulaF  
 1, 2, 52, 53  
 InsulaG  
 61, 62  
 InsulaX  
 3, 5, 7, 8

**OBS:**

- In acest caz nu se mai ia in calcul distribuirea tuturor posibilitatilor (se poate observa ca insulaG nu a primit niciun avion).
- Au prioritate insulele care pot asigura numarul de locuri egal cu numarul de avioane cedate de insula Sursa  
 exemplu:  
 InsulaX cedeaza 2 avioane si are conexiune doar cu insulaY (aceasta are toate locurile ocupate) care are urmatoarele conexiuni:  
 InsulaA poate sa primeasca 1 avion  
 InsulaB poate sa primeasca 2 avioane  
 InsulaC poate sa primeasca 2 avioane  
 Doar insulaB o sa primeasca 2 avioane de la insulaY  
 //nu se mai tine cont de posibilitatile de distribuire => se face doar una singura
- Daca toate insulele cu care are conexiune InsulaX nu au cel putin numarul de locuri necesar, se reia procesul pornind de la prima insula cu care insulaX are legatura.  
 Adica, se efectueaza cautari pana cand se gaseste o insula care poate sa primeasca numarul de avioane cedate. Daca nu exista o astfel de insula, dar in arhipelag numarul total de locuri disponibile este egal cu numarul de avioane cedate => avioanele o sa fie distribuite in acele locuri. Daca nu exista o astfel de insula si in arhipelag nu exista un numar total de locuri disponibile egal cu numarul de avioane cedate => Stack overflow!

7) Mai multe insule distribuie avioane, dar insulele cu care sunt conectate nu au capacitate sa le primeasca: Combinatie intre 5 si 6

**IMPORTANT:** Testele 9 - 14 au in prima parte a fisierului de date.in inserate datele de la testul 1. Pentru a putea rezolva etapa de backtracking este necesar sa rezolvati primul test.

**Restrictii:**

Id-urile avioanelor de pe fiecare insula trebuie sa fie ordonate crescator. Id-urile avioanelor au valori intre 1 si 100.

In cazul in care, o insula depaseste toleranta, pe insula vor ramane avioanele cu id-urile cele mai mici si vor fi redistribuite avioanele cu id-urile cele mai mari din lista:

InsulaX are avioanele cu id-urile 3, 1, 6 12, 2 si 19. Toleranta e 3.

Pe insula trebuie sa ramana avioanele cu id-urile 1, 2 si 3. Se vor redistribui avioanele cu id-urile 6, 12 si 19. Id-urile mai mari au mereu prioritate.

### **EXEMPLU BACKTRACKING** (folosim datele prezentate mai sus)

#### ***Intelegerea datelor:***

Prelucrând fisierul de date.in, constatam ca pentru aceasta etapa avem:

Numar de insule: 5 //o sa ii spunem n

Toleranta: 4

Un vector care reprezinta numarul de avioane pentru fiecare insula:

1 6 2 1 3

Observatie:

- $v[0]$  este insula1,  $v[1]$  este insula[2] etc
- $v[0] = 1 \Rightarrow$  insula1 are un avion
- $v[1] = 6 \Rightarrow$  insula2 are 6 avioane

Apoi avem n linii care reprezinta vectorul de avioane pentru fiecare insula:

1 //insula1

3 9 17 18 22 98 //insula2

4 7 //insula3

12 //insula4

19 21 23 //insula5

Dupa aceea, avem matricea de adiacenta care ne indica intre ce insule putem face distribuirea:

0 1 0 0 1

1 0 1 1 0

0 1 0 1 0

0 1 1 0 0

1 0 0 0 0

#### ***Cum trebuie sa manipulam aceste date?***

Observam ca insula2 a depasit toleranta, avand un surplus pe 2 avioane pe care trebuie sa le distribuie insulelor cu care este conectata (insula1, insula3, insula4). Asadar, trebuie sa cedeze avioanele cu id-ul 22 si 98.

Prima insula care primeste avioane este insula1, deci o sa avem urmatoarele posibilitati:

- insula1 primeste 2 avioane [22 si 98]
- insula1 primeste 1 avion [98]
- insula1 primeste 1 avion [22]
- insula1 primeste 0 avioane [se afiseaza doar avioanele de pe insula]

Observatie: Insula1 poate sa primeasca 0 avioane deoarece cele 2 avioane pot fi distribuite celorlalte insule cu care insula2 se afla in contact (pot sa accepte numarul maxim de avioane cedate - individual sau impreuna).

Urmarind aceeasi logica si pentru urmatoarele insule, reiese ca fisierul de rezultate.out ar arata astfel:

OK

NO //cerintele de la etapa de analiza

Island1 //partea de distributie

1 22 98 //poate primi 2 avioane

1 98 //poate primi 1 avion

1 22 //poate primi 1 avion

1 //poate primi 0 avioane

Island2

3 9 17 18 //a cedat cele 2 avioane

Island3  
4 7 22 98 //poate primi 2 avioane  
4 7 98 //poate primi 1 avion  
4 7 22 ///poate primi 1 avion  
4 7 //poate primi 0 avioane  
Island4  
12 22 98 //poate primi 2 avioane  
12 98 //poate primi 1 avion  
12 22 //poate primi 1 avion  
12 //poate primi 0 avioane  
Island5  
19 21 23 //nu a primit avioane

Observatie: Ce facem in cazul in care avem de cedat 3 avioane? (luam ca referinta matricea de adiacenta prezentata mai sus)

Exemplu: trebuie sa cedam avioanele cu idurile [22, 55, 77]

Ce cazuri posibile avem?

InsulaX primeste 3 avioane: 22 55 77

InsulaX primeste 2 avioane: 55 77

InsulaX primeste 2: avioane 22 55

InsulaX primeste 1 avion: 77

InsulaX primeste 1 avion: 55

InsulaX primeste 1 avion: 22

InsulaX primeste 0 avioane

IMPORTANT: De ce nu avem varianta InsulaX primeste 2 avioane: 22 si 77? Deoarece distribuim prima oara avioanele cu id-ul maxim.

InsulaX primeste 2 avioane, iar insulaY primeste 1 avion:

Deci, insulaX poate sa primeasca 55 77 si insulaY 22 sau InsulaY primeste 77 si atunci insulaX primeste 22 55.

Observatie: se vor folosi obligatoriu **alocari dinamice si eliberarile de memorie aferente**, in caz contrar se vor aplica depunctari.

### Structuri de date necesare

#### Structura pentru resurse:

```
typedef struct Resursa {  
    char *nume;  
    int cantitate;  
} Resursa;
```

#### Structura pentru insula:

```
typedef struct Island{  
    char *nume;  
    int nrResurse;  
    Resursa *inventarResurse;  
    int nrAvioane;  
    int *avioane;  
    int tolAvioane;  
} Island;
```

### ***Punctaj***

Test 1: 2 pcte

Test 2: 3 pcte

Test 3-4: 5 pcte / test =  $2 \times 5 = 10$  pcte

Test 5-7: 15 pcte / test =  $3 \times 15 = 45$  pcte

Test 8: 20 pcte

Test 9: 6 pcte

Test 10: 4 pcte

Test 11-12: 10 pcte / test = 20 pcte

Test 13: 10 pcte

Test 14: 10 pcte

Makefile: 10 pcte

Readme: 10 pcte

### ***Incarcare tema***

Fisierele care contin rezolvarea temei se incarca sub forma de arhiva cu numele Nume\_Prenume\_Grupa.\* pe Moodle in contul lucrarii TEMA2.

Cand arhivati, fisierele se plaseaza direct in radacina! Nu faceti un director in care se afla toate sursele si arhivati directorul, ci selectati toate sursele si faceti arhiva din ele.