

## Tema 2 –Metoda Greedy

1. Se dau N spectacole, pentru fiecare cunoscându-se identificatorul, ora de început și ora de sfârșit a spectacolului (ore care definesc intervalul de timp în care sala este rezervată, asigurându-se și intrarea/ieșirea spectatorilor) Să se determine un număr maxim de spectacole care pot fi programate în aceeași sală și în aceeași zi.

**Date de intrare:** fișierul date.in conține:

- pe prima linie un număr natural N, cu semnificația din enunț;
- pe următoarele N linii triplete de numere naturale de forma id start stop, unde id – identificatorul unui spectacol, start – ora de la care se poate intra în sală pentru acest spectacol, stop – ora la care s-a eliberat sala după acest spectacol.

**Date de ieșire:** fișierul date.out conține identificatorii spectacolelor selectate, în ordinea programării lor, câte unul pe rând.

**Exemplu:**

| date.in | date.out |
|---------|----------|
| 5       | 2        |
| 1 9 11  | 5        |
| 2 8 10  | 3        |
| 3 12 14 |          |
| 4 7 13  |          |
| 5 10 12 |          |

2. În N scorbură, numerotate de la 1 la N, se găsesc câte  $A_1, A_2, \dots, A_N$  alune ( $A_i < A_{i+1}$ ). O veveriță are M chitanțe pe care trebuie să le distribuie la M scorbură. Pe fiecare chitanță i este trecut suma  $S_i$ , dată sau primită pentru o alună, precum și tipul tranzacției  $T_i$  (-1 – sumă dată sau +1 pentru sumă primită). În funcție de chitanța dată la o scorbură, veverița fie primește suma corespunzătoare tuturor alunelor din aceasta, fie plătește suma corespunzătoare tuturor alunelor din aceasta. Scrieți programul care ajută veverița să distribuie aceste chitanțe celor N scorbură astfel încât suma totală cu care rămâne să fie maximă sau datoriile să fie minime.

**Date de intrare:** fișierul date.in conține:

- prima linie două numere naturale,  $N$  și  $M$ , cu semnificația din enunț
- pe linia a doua  $N$  numere naturale,  $A_1 \dots A_N$ , cu semnificația din enunț
- pe următoarele  $M$  linii, triplete de numere de forma  $i, S_i, T_i$ , unde  $i$  este id-ul unei chitanțe, iar  $S_i$  și  $T_i$  au semnificația din enunț ( $i$  și  $S_i$  sunt numere naturale, iar  $T_i$  număr întreg)

**Date de ieșire:** fișierul date.out conține suma totală cu care rămâne veverița.

### Exemplu:

Exemplu – o soluție poate fi:

| date.in   | date.out | Explicație   |
|---|----------|--|
| 6 4<br>1 3 4 6<br>7 9<br>1 2 -1<br>2 4 1<br>3 3 1<br>4 5 -1 | 46       | Au fost distribuite chitanțe astfel: 1 la scorbura 2, 2 la scorbura 6, 3 la scorbura 5, 4 la scorbura 1. Suma totală obținută în urma tranzacțiilor este 46<br>( $46 = -2 \cdot 3 + 4 \cdot 9 + 3 \cdot 7 - 5 \cdot 1$ ) |

3. Pentru o scorbură din pădure, veverițele depun  $N$  cereri de cazare, numerotate de la 1 la  $N$ . În fiecare cerere “ $i$ ” se specifică ziua din luna curentă ( $1, 2, \dots, 30$ ) în care începe sejurul,  $S_i$ , precum și numărul de zile ale acestuia,  $Z_i$ , astfel încât sejurul are loc în totalitate doar pe parcursul lunii curente. Știind că într-o zi nu pot fi cazate două veverițe în aceeași scorbură, scrieți programul care determină numărul maxim de cereri care pot primi răspuns favorabil, precum și numărul de ordine al acestora.

**Date de intrare:** fișierul date.in conține:

- prima linie  $N$ , cu semnificația din enunț
- pe următoarele  $N$  linii, triplete de numere naturale de forma  $i, S_i, Z_i$ , unde  $i$  este id-ul unei cereri, iar  $S_i$  și  $Z_i$  au semnificația din enunț

**Date de ieșire:** fișierul date.out conține

- pe prima linie, numerele corespunzătoare cererilor, separate prin câte un spațiu
- pe a doua linie numărul total de zile în care scorbura a fost ocupată

### Exemplu:

| date.in   | date.out      | Explicație   |
|---|---------------|--|
| 6<br>1 4 2<br>2 1 3<br>3 6 2<br>4 3 4<br>5 9 5<br>6 7 4 | 1 2 3 5<br>12 | Au fost selectate cererile 1 (care ocupă scorbura în zilele de 4 și 5), 2 (care ocupă scorbura în zilele de 1,2 și 3), 3 (care ocupă scorbura în zilele de 6 și 7), 5 (care ocupă scorbura în zilele de 9,10,11,12 și 13). |

4. Se dau N intervale închise și se cere reuniunea acestora.

**Date de intrare:** fișierul date.in conține:

- pe prima linie un număr natural N, cu semnificația din enunț;
- pe următoarele N linii perechi de numere naturale de forma x y, unde unde x și y reprezintă capetele unui interval închis [x,y].

**Date de ieșire:** fișierul date.out conține capetele intervalelor rezultate prin reuniune, câte o pereche pe rând; numerele de pe același rând sunt separate prin câte un spațiu.

**Restricții și precizări:**

$N \in [2, 10^2]$

Pentru fiecare interval

$x, y \in [1, 10^9], x \leq y$

**Exemplu:**

| date.in   | date.out              |
|---|-----------------------|
| 6<br>80 85<br>3 7<br>50 70<br>83 84<br>1 5<br>25 50 | 1 7<br>25 70<br>80 85 |

5. Se dau  $N$  intervale închise. Să se determine un număr minim de puncte care se pot alege pe axa  $Ox$  astfel încât orice interval să conțină cel puțin un punct dintre acestea (dacă sunt mai multe soluții, se alege doar una dintre ele).

**Date de intrare:** fișierul `date.in` conține:

- pe prima linie un număr natural  $N$ , cu semnificația din enunț;
- pe următoarele  $N$  linii perechi de numere naturale de forma  $x\ y$ , unde unde  $x$  și  $y$  reprezintă capetele unui interval închis  $[x,y]$ .

**Date de ieșire:** fișierul `date.out` conține abscisele punctelor cerute, câte unul pe rând.

**Restricții și precizări:**

$N \in [2, 10^2]$

Pentru fiecare interval

$x, y \in [1, 10^9], x \leq y$

**Exemplu:**

| date.in | date.out |
|---------|----------|
| 6       | 4        |
| 80 85   | 50       |
| 3 7     | 83       |
| 50 70   |          |
| 83 84   |          |
| 1 5     |          |
| 25 50   |          |

6. Se da o mulțime de  $N$  numere întregi distincte și un număr  $k$ . Să se determine o submulțime cu număr maxim de elemente a mulțimii date, astfel încât în aceasta să fie cel mult  $k$  numere prime, a caror sumă să fie strict mai mare decât suma numerelor selectate care nu sunt prime.

**Date de intrare:** fișierul `date.in` conține:

- pe prima linie două numere naturale  $N$  și  $k$ , cu semnificația din enunț;
- pe următoarele  $N$  linii numere întregi, reprezentând elementele mulțimii.

**Date de ieșire:** fișierul `date.out` conține elementele submulțimii.

**Exemplu:**

| date.in                        | date.out      | Explicatie        |
|--------------------------------|---------------|-------------------|
| 11 3<br>3 2 1 9 5 7 8 6 4 12 0 | 0 1 3 4 5 6 7 | $3+5+7 > 0+1+4+6$ |

7. Se dau N intervale (încj=hi se la primul capăt și deschise la celalalt capăt) și se cere să se aleagă un număr maxim de intervale astfel încât oricare două intervale alese să nu se intersecteze.

**Date de intrare:** fișierul date.in conține:

- pe prima linie un număr natural N, cu semnificația din enunț;
- pe următoarele N linii perechi de numere naturale de forma x y, unde x și y reprezintă capetele unui interval [x,y).

**Date de ieșire:** fișierul date.out conține, pe linii separate, câte o pereche de valori reprezentând capetele unui interval ales

**Exemplu:**

| date.in | date.out |
|---------|----------|
| 6       | 4 8      |
| 8 10    | 8 10     |
| 2 20    | 12 16    |
| 4 8     | 17 18    |
| 12 16   |          |
| 3 18    |          |
| 17 18   |          |

8. O asociatie caritabila asigura consultatii medicale gratuite, dar exista un singur cabinet dotat cu aparatura medicala. Din acest motiv la un moment dat un singur medic poate face consultatii. Asociatia apeleaza la N medici de diverse specialitati, care isi ofera benevol serviciile. Fiecare prezinta un singur interval [si,fi] de-a lungul aceleiasi zile, in care este disponibil. Ajutati asociatia sa realizeze o planificare a consultatiilor in cabinet, astfel incat numarul de medici sa fie maxim.

**Date de intrare:** fișierul date.in conține:

- pe prima linie un număr natural N, cu semnificația din enunț;

- pe următoarele N linii triplete de numere naturale de forma  $i$   $s_i$   $f_i$ , unde  $i$  este identificatorul medicului, iar  $s_i$   $f_i$  au semnificația din enunț

**Date de ieșire:** fișierul date.out conține identificatorii medicilor selectați, în ordinea programării lor, câte unul pe rând.

**Exemplu:**

| date.in | date.out |
|---------|----------|
| 5       | 2        |
| 1 9 11  | 5        |
| 2 8 10  | 3        |
| 3 12 14 |          |
| 4 7 13  |          |
| 5 10 12 |          |

9.

Se dau două mulțimi de numere întregi,  $A$  și  $B$ , cu  $NA$ , respectiv  $NB$  elemente. Se cere să se determine o submulțime  $X$  a lui  $A$  astfel încât expresia  $x_1 * b_1 + x_2 * b_2 + \dots + x_{nb} * b_{nb}$  să aibă o valoare maximă.

**Date de intrare:** fișierul date.in conține:

- pe prima linie două numere naturale  $NA$  și  $NB$ , cu semnificația din enunț;
  - pe a doua linie  $NA$  numere întregi, reprezentând elementele mulțimii  $A$ ;
  - pe a treia linie  $NB$  numere întregi, reprezentând elementele mulțimii  $B$ .
- Elementele de pe aceeași linie sunt separate prin câte un spațiu.

**Date de ieșire:** fișierul date.out valoarea expresiei determinate.

**Exemplu:**

| date.in     | date.out |
|-------------|----------|
| 5 3         | 51       |
| 5 -3 8 -1 2 |          |
| 2 4 3       |          |

10. La o expoziție există  $N$  mașini disponibile pentru test-drive. Ionel vrea să conducă (testeze) cât mai multe dintre aceste mașini. Având orarul în care aceste mașini sunt disponibile ( $s_i$ ,  $f_i$  – pentru fiecare mașină  $i$ ) ajutați-l pe Ionel să își rezerve cât mai multe mașini pentru test-drive.

**Date de intrare:** fișierul date.in conține:

- pe prima linie un număr natural  $N$ , cu semnificația din enunț;
- pe următoarele  $N$  linii triplete de numere naturale de forma  $i \ s_i \ f_i$ , unde  $i$  este identificatorul mașinii, iar  $s_i \ f_i$  au semnificația din enunț

**Date de ieșire:** fișierul `date.out` conține identificatorii mașinilor selectate, în ordinea programării lor, câte unul pe rând.

**Exemplu:**

| date.in | date.out |
|---------|----------|
| 5       | 2        |
| 1 9 11  | 5        |
| 2 8 10  | 3        |
| 3 12 14 |          |
| 4 7 13  |          |
| 5 10 12 |          |

**11.** Se dau  $N$  reactivi. Fiecare reactiv  $i$  se păstrează în frigider între temperaturile  $[Ti1, Ti2]$ . Se cere numărul minim de frigidere necesare știind că mai mulți reactivi pot fi plasați în același frigider dacă temperatura acestuia o permite.

**Date de intrare:** fișierul `date.in` conține:

- pe prima linie un număr natural  $N$ , cu semnificația din enunț;
- pe următoarele  $N$  linii perechi de numere întregi de forma  $Ti1 \ Ti2$ , cu semnificația din enunț.

**Date de ieșire:** fișierul `date.out` conține numărul cerut.

**Exemplu:**

| date.in | date.out |
|---------|----------|
| 5       | 3        |
| 2 5     |          |
| 3 4     |          |
| 2 3     |          |
| 4 7     |          |
| 6 8     |          |

**12.** Un bijutier are la dispoziție  $N$  bucăți de metal prețios. Pentru fiecare bucată  $i$  de metal el cunoaște greutatea  $g_i$  (în grame) și prețul  $p_i$  (în \$) . Din aceste metale el trebuie să confecționeze un colier împletit, având greutatea  $G$  impusă de către cumpărător. Știind că bijuteria trebuie realizată cu un cost minim, să se precizeze cum folosește bijutierul bucățile de metal prețios.  
Obs. Bucățile de metal prețios pot fi folosite în întregime sau numai parțial.

**Date de intrare:** fișierul `date.in` conține:

- pe prima linie două numere naturale  $N$  și  $G$ , cu semnificația din enunț;
- pe următoarele  $N$  linii triplete de numere naturale de forma  $id\ g\ p$ , unde  $id$  reprezintă identificatorul unui metal,  $g$  greutatea acestuia, iar  $p$  prețul sau integral.

**Date de ieșire:** fișierul `date.out` conține pe fiecare linie identificatorul unui metal selectat și procentul în care au fost inclus în bijuterie. Pe ultima linie se scrie valoarea totală a acesteia.

**Exemplu:**

| date.in | date.out |
|---------|----------|
| 4 10    | 1 100%   |
| 1 2 3   | 4 100%   |
| 2 6 10  | 2 0.16%  |
| 3 3 15  | 16.26    |
| 4 8 13  |          |

**13.** Se citesc de la tastatura două numere naturale  $n$  și  $k$ . Generați o matrice cu  $n$  linii și  $n$  coloane, care să îndeplinească simultan următoarele condiții:

- conține toate numerele naturale de la 1 la  $n^2$ , fiecare număr luat o singură dată
- pe fiecare linie elementele sunt așezate crescător de la stanga la dreapta
- suma elementelor de pe o coloană  $k$  să fie minimă