

METODA GREEDY (metoda optimului local)**I. Repere teoretice**

- Se aplică problemelor de optim – dintr-o mulțime de elemente A se cere o submulțime B care verifică anumite condiții (de obicei soluția unei probleme de optimizare)
- Soluția se construiește pe măsura parcurgerii datelor:
 - Inițial mulțimea B este vidă
 - La fiecare pas se alege cel mai promițător element din A care ar putea conduce la o soluție
 - Se elimină elementul ales din A și, dacă el conduce la o soluție, se include în B;
- De multe ori este utilă ordonarea mulțimii A astfel încât elementele cele mai promițătoare să apară primele:
 - Inițial mulțimea B este vidă
 - Se prelucrează/ordonează mulțimea A
 - Se parcurge mulțimea A ordonată: dacă a_i poate conduce la o soluție (nu neapărat optimă), se adaugă la B
- Observații:
 - optimul global se determină prin estimări succesive ale optimului local
 - nu se revine asupra deciziilor făcute: dacă un element a_i a fost introdus în B, el nu va mai fi eliminat din aceasta, iar dacă un element a_i nu a fost introdus în B, el nu va mai fi testat ulterior.
- Metoda nu oferă întotdeauna soluția optima (greedy euristic) – optimalitatea soluțiilor problemelor rezolvate cu metoda greedy trebuie demonstrate matematic (de obicei prin inducție matematică sau prin reducere la absurd).

1. Fișierul date.in conține cel mult 10000 de numere întregi, reprezentând elementele unei mulțimi.

Să se determine o submulțime S a mulțimii date, astfel încât suma elementelor lui S să fie maximă. Datele se scriu în fișierul date.out.

(Indicație: se aleg toate elementele pozitive ale mulțimii)

2. Fișierul date.in conține pe prima linie numerele n și k ($k \leq n \leq 10000$), apoi n numere întregi, reprezentând elementele unei mulțimi.

Să se determine o submulțime S a mulțimii date, astfel încât aceasta să aibă cardinalul k și suma elementelor lui S să fie maximă. Datele se scriu în fișierul date.out.

(Indicație: se ordonează mulțimea și se aleg cele mai mari k elemente ale acesteia)

3. Problema spectacolelor

Fișierul date.in conține pe prima linie un număr natural n ($n \leq 10000$), iar pe următoarele n linii triplete 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.

Să se determine un număr maxim de spectacole care pot fi programate în aceeași sală și în aceeași zi. Identificatorii spectacolelor selectate se scriu în fișierul date.out, în ordinea programării lor.

Exemplu:

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

(Indicație: se ordonează datele după ora de eliberare a sălii de spectacole, astfel încât timpul rămas disponibil după alegerea unui spectacol să fie cât mai mare)

4. Reuniunea intervalelor

Fișierul date.in conține pe prima linie un număr natural n ($n \leq 10000$), iar pe următoarele n linii perechi de forma x y, unde x și y reprezintă capetele unui interval închis $[x,y]$.

Să se determine reuniunea acestor intervale. Datele se scriu în fișierul date.out.

date.in	date.out
6 80 85	[1,7] U [25,70] U [80,85]

3 7 50 70 83 84 1 5 25 50	
---------------------------------------	--

(Indicație: se ordonează datele după limita din stânga a intervalelor, deoarece în cadrul reuniunii primele intervale care apar au capătul din stânga minim)

5. Puncte pe axă

Fișierul date.in conține pe prima linie un număr natural n ($n \leq 10000$), iar pe următoarele n linii perechi de forma $x y$, unde x și y reprezintă capetele unui interval închis $[x,y]$. 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 acestea).

Datele se scriu în fișierul date.out.

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

(Indicație: se ordonează datele după limita din stânga a intervalelor și se determină intervalele de intersecție a celor alăturate în mulțimea ordonată, din care se alege câte un punct)

6. Problema rucsacului (varianta continuă)

Fișierul date.in conține pe prima linie două numere naturale n și G ($n \leq 10000$), iar pe următoarele n linii triplete de forma $k v c$, unde k reprezintă identificatorul unui obiect, v este valoarea acestuia, iar c cantitatea sa.

Să se determine obiectele care pot fi încărcate într-un rucsac care poate transporta o greutate maximă G , astfel încât valoarea totală a acestora să fie maximă. Obiectele nu trebuie neapărat transportate integral (de exemplu se poate alege doar 1kg de făină și nu întreaga cantitate disponibilă). Identificatorii obiectelor selectate precum și valoarea totală a acestora se scriu în fișierul date.out.

date.in	date.out
4 10 1 2 3 2 6 10 3 3 15 4 8 13	2 3 4 (partial) 26.625

(Indicație: se ordonează obiectele după valoarea lor unitară)

7. Șir de numere

Fișierul date.in conține pe prima linie două numere naturale n și k ($k < n \leq 10000$), iar pe a doua linie un șir de n numere întregi.

Se definește următoarea operație: se poate schimba semnul unei secvențe de k numere din șir începând cu orice poziție i ($1 \leq i \leq n-k+1$)

Să se determine, dacă este posibil, o succesiune de operații aplicate șirului în poziții convenabile astfel încât la final toate elementele șirului să fie pozitive și numărul operațiilor să fie minim. Pozițiile în care se aplică operația se scriu în fișierul date.out.

date.in	date.out
7 3 1 2 -3 -1 5 -2 -1	2 4

(Indicație: se parcurge șirul între pozițiile $0..n-k$ și dacă pe o astfel de poziție se află un număr negativ, se aplică operația)

8. Interclasare

Fișierul date.in conține pe prima linie un număr natural n ($n \leq 10000$), iar pe a doua linie un șir de n numere naturale, reprezentând lungimile a n șiruri ordonate crescător.

Se știe că dacă se interclasează două șiruri de lungime x și y se fac $x+y$ operații și se obține un șir de lungime $x+y$. Să se determine o modalitate de a interclasa cele n șiruri astfel încât să se obțină un singur șir, iar numărul de operații să fie minim.

Identificatorii șirurilor care se interclasează se scriu în fișierul date.out, în ordinea efectuării operației de interclasare.

date.in	date.out
5	1 3

2 7 1 6 4	(1-3) 5 2 4 (1-3-5) (2-4)
-----------	---------------------------------

(Indicație: se selectează la fiecare pas câte două șiruri de dimensiune minimă și se interclasează, obținându-se un nou șir)

9. Expresie de sumă maximă

Fișierul date.in conține pe prima linie două numere naturale n_a și n_b ($n_b \leq n_a \leq 10000$), pe a doua linie n_a numere întregi, reprezentând elementele mulțimii A, iar pe a treia linie n_b numere întregi, reprezentând elementele mulțimii B.

Să se determine o submulțime X a lui A astfel încât expresia $x_1 \cdot b_1 + x_2 \cdot b_2 + \dots + x_{n_b} \cdot b_{n_b}$ să aibă o valoare maximă.

Valoarea expresiei se scrie în fișierul date.out.

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

(Indicație: se ordonează mulțimile crescător și se tratează cele trei cazuri posibile: mulțimea A are toate elementele pozitive, toate elementele negative sau este mixtă)

10. Problema votului majoritar

Se dă un șir de n numere naturale. Se cere determinarea unui element care apare de cel puțin $\lfloor n/2 \rfloor + 1$ ori în șir dacă există un astfel de element în șir. Dacă exista se afiseaza DA si numarul respectiv, altfel se afiseaza NU.

date.in	date.out
8 2 2 3 4 3 3 7 3	DA 3

(Indicație: se parcurge lista votanților, luându-se în considerare votantii unui potential candidat: se numără voturile în neîmperecheate (unul la candidat+altul la opoziție se anulează reciproc) – dacă votul curent îl susține pe cel potential, contorul crește, iar dacă este din opoziție, contorul scade; dacă toate voturile sunt anulate reciproc (contor=0), atunci apare un nou potential candidat. Cel rămas în final este castigator, dacă nr voturilor este $> n/2$).