Elaborarea algoritmilor

Metoda GREEDY

Metoda Greedy

- **Probleme de optim**
- **Cadru**:

  Se dă o mulțime finită A.

  Să se determine o submulțime finită B⊆A care satisface anumite condiții (este **soluție posibilă**)

  $+$

  îndeplinește un criteriu de optim (este **soluție optimă**)

Metoda Greedy

- **Probleme de optim**
- **Cadru**:

  Se dă o mulțime finită A.

  Să se determine o submulțime finită $B \subseteq A$ care satisface anumite condiții (este **soluție posibilă**)

  $+$

  îndeplinește un criteriu de optim (este **soluție optimă**)

- **Soluțiile posibile au proprietățile**
  - $\emptyset$ este soluție posibilă
  - Dacă B este soluție posibilă și $C \subseteq B$, atunci C este soluție posibilă

Metoda Greedy

- **Idee**: Se încearcă o construire directă a **<u>unei</u>** soluții optime, **<u>element cu element</u>**
- Elementul ales la un pas pentru a se adăuga în soluție – ***cel care pare cel mai "bun" la acel pas***, conform criteriului de optim

Metoda Greedy

- **Idee**: Se încearcă o construire directă a **<u>unei</u>** soluții optime, **<u>element cu element</u>**

- Elementul ales la un pas pentru a se adăuga în soluție – ***cel care pare cel mai "bun" la acel pas***, conform criteriului de optim

- Nu este garantată obținerea unei soluții optime ⇒ **aplicarea metodei trebuie însoțită de demonstrația corectitudinii**

Metoda Greedy

- **Cadru formal**: Se dau
  - O mulțime finită $A=\{a_1,...,a_n\}$
  - O proprietate definită pe mulțimea submulțimilor lui A

$$p:\mathcal{P}(A) \rightarrow \{0,1\} \text{ astfel încât}$$

$$\begin{cases} p(\phi)=1 \\ p(X) \Rightarrow p(Y),\ \forall Y \subset X \end{cases}$$

Metoda Greedy

- **Cadru formal**: Se dau
  - O mulțime finită $A=\{a_1,...,a_n\}$
  - O proprietate definită pe mulțimea submulțimilor lui A

$$p: \mathcal{P}(A) \rightarrow \{0,1\} \text{ astfel încât}$$

$$\begin{cases} p(\phi)=1 \\ p(X) \Rightarrow p(Y), \forall Y \subset X \end{cases}$$

  - O funcție $f: \mathcal{P}(A) \rightarrow \mathbf{R}$

Metoda Greedy

- **Cadru formal**: Se dau
  - O mulțime finită $A=\{a_1,\dots,a_n\}$
  - O proprietate definită pe mulțimea submulțimilor lui A

    $p: \mathcal{P}(A) \rightarrow \{0,1\}$ astfel încât

$$\begin{cases} p(\phi)=1 \\ p(X) \Rightarrow p(Y),\ \forall Y \subset X \end{cases}$$

  - O funcție $f: \mathcal{P}(A) \rightarrow \mathbf{R}$

O submulțime $S \subset A$ sn **<u>soluție posibilă</u>** dacă $p(S)=1$.

Metoda Greedy

- **Cadru formal**: Se dau
  - O mulțime finită $A=\{a_1,\ldots,a_n\}$
  - O proprietate definită pe mulțimea submulțimilor lui A

    $p:\mathcal{P}(A) \rightarrow \{0,1\}$ astfel încât

$$\begin{cases} p(\phi)=1 \\ p(X) \Rightarrow p(Y),\ \forall Y \subset X \end{cases}$$

  - O funcție $f:\mathcal{P}(A)\rightarrow \mathbf{R}$

O submulțime $S \subset A$ sn **<u>soluție posibilă</u>** dacă $p(S)=1$.

Să se determine o soluție posibilă care optimizează funcția f

Metoda Greedy

- **Două variante** (modalități de abordare)

  - Varianta 1 – **fără prelucrare inițială**: **elementul** care se adaugă la soluție se stabilește **la fiecare pas**, în funcție de alegerile anterioare

  - Varianta 2 – **cu prelucrare inițială:** ordinea în care sunt considerate elementele se stabilește de la început

Exemple

Probleme de planificare activități cu intervale de desfășurare date

Problema spectacolelor (selecția unui număr maxim de intervale disjuncte)

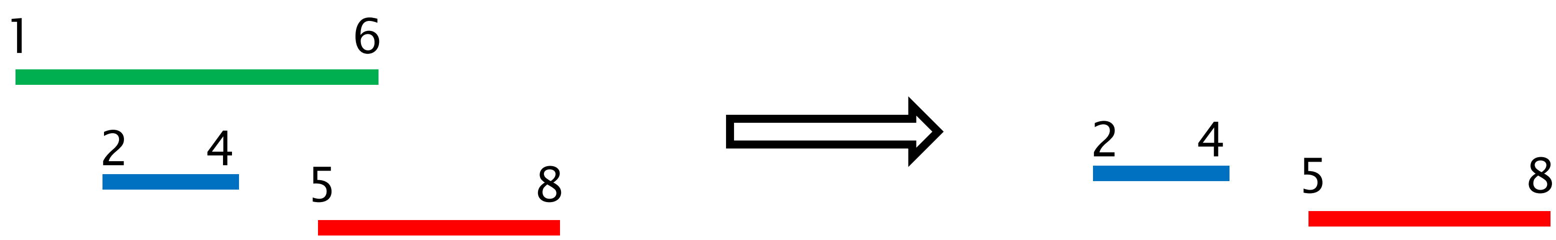
Se dă o mulțime de n intervale disjuncte (reprezentând intervalele de desfășurare a n activități, spre exemplu spectacole)

Să se determine o submulțime de cardinal maxim de intervale disjuncte două câte două (de activități compatibile, care se pot desfășura folosind o singură resursă)

Probleme de planificare activități cu intervale de desfășurare date

Problema spectacolelor (selecția unui număr maxim de intervale disjuncte)

Exemplu

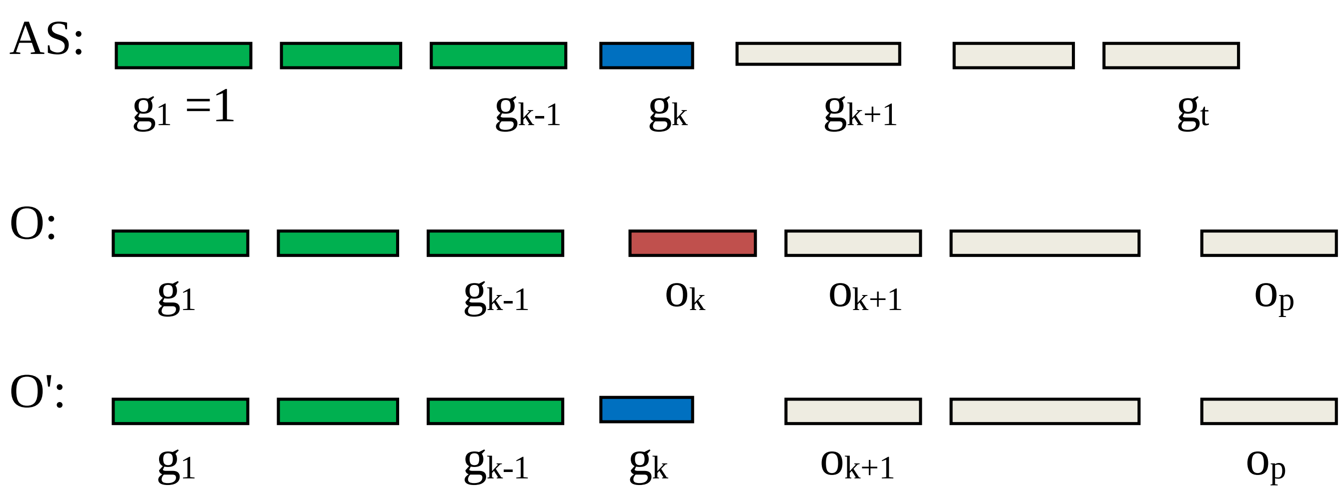


Soluție optimă

Probleme de planificare activități cu intervale de desfășurare date

Problema spectacolelor (selecția unui număr maxim de intervale disjuncte)

Soluție + corectitudine – la curs + pdf greedy



Probleme de planificare activități cu intervale de desfășurare date

Problema spectacolelor (selecția unui număr maxim de intervale disjuncte)

Variante

- Activitățile au asociate și **profituri**. Să se determine o submulțime de activități compatibile având profitul total maxim (un algoritm de tip Greedy similar celui pentru selectarea submulțimii de cardinal maxim nu mai este corect)

Probleme de planificare activități cu intervale de desfășurare date

Problema spectacolelor (selecția unui număr maxim de intervale disjuncte)

Variante

- **Problema partiționării intervalelor**: De câte resurse este nevoie pentru a putea planifica toate activitățile date + o astfel de planificare (tema laborator Greedy)

Minimizarea timpului mediu de acces pentru acces secvențial

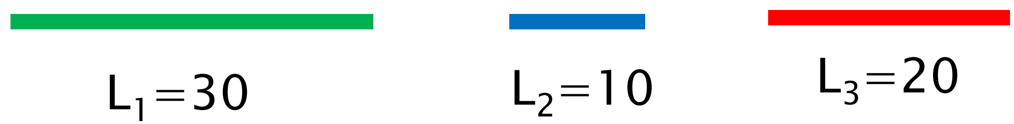
- n texte cu lungimile L(1),...,L(n) urmează a fi așezate pe o bandă cu acces secvențial

  Acces secvențial = pentru a citi textul de pe poziția k, trebuie citite textele de pe pozițiile 1,2,...,k

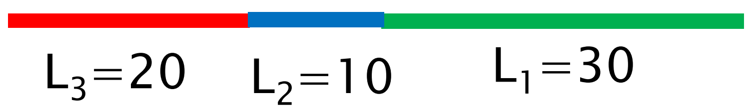
  Să se determine o modalitate de așezare a textelor pe bandă astfel încât timpul mediu de acces să fie minimizat.

Minimizarea timpului mediu de acces pentru acces secvențial

- Exemplu



Pentru așezarea



timpul mediu de acces este...

Minimizarea timpului mediu de acces pentru acces secvențial

- **Variantă de enunț:** Un evaluator de proiecte (bucătar) urmează să evalueze n proiecte (să efectueze n comenzi) depuse de directorii de proiect. Pentru fiecare proiect se știe durata de evaluare.

  Care este ordinea în care se va face evaluarea astfel încât să se minimizeze timpul total de așteptare al directorilor de proiect (clienților)?

  - Răspunsul este dat celui care a depus proiectul imediat după terminarea evaluării acestuia (nu a tuturor proiectelor)

Minimizarea timpului mediu de acces pentru acces secvențial

- Reprezentarea soluției

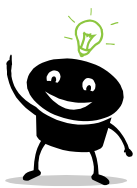
- Timpul mediu de acces (funcția de optimizat)

Minimizarea timpului mediu de acces pentru acces secvențial

Care este primul text pe care îl așezăm pe bandă?

Minimizarea timpului mediu de acces pentru acces secvențial

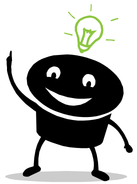
Care este primul text pe care îl așezăm pe bandă?



Cel cu lungimea cea mai mică, pentru că el va fi accesat de câte ori accesăm un alt text

Minimizarea timpului mediu de acces pentru acces secvențial

Care este primul text pe care îl așezăm pe bandă?



Cel cu lungimea cea mai mică, pentru că el va fi accesat de câte ori accesăm un alt text

<u>**Soluție Greedy**</u>: Așezăm textele pe bandă în **ordine crescătoare** în raport cu lungimea lor

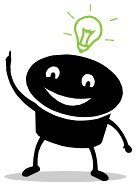
Complexitate O(nlogn)

Minimizarea timpului mediu de acces pentru acces secvențial

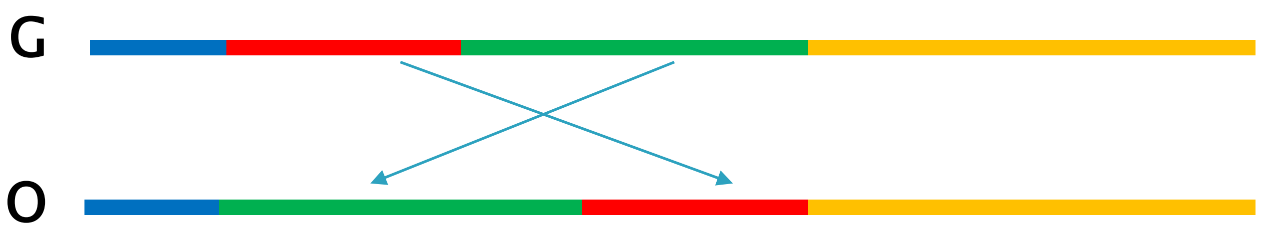
Este corect algoritmul?

Minimizarea timpului mediu de acces pentru acces secvențial

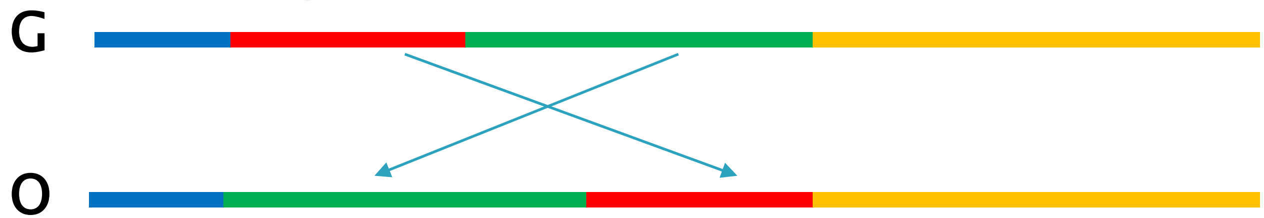
Este corect algoritmul?



Comparăm soluția dată de algoritmul de tip Greedy cu una optimă



Minimizarea timpului mediu de acces pentru acces secvențial



Greedy – permutarea identică

Soluția optimă – permutarea p – are minim o inversiune

⇒ o nouă permutare p’ cu mai puține inversiuni

p: ...., p_i ,............., p_j ,........

p: ...., p_j ,............., p_i ,........

p’ este optimă?

Detalii – la curs + pdf greedy

Minimizarea timpului mediu de acces pentru acces secvențial

- **Alte tipuri de probleme**

1. Fiecare text are asociată o frecvență de citire
2. Avem la dispoziție p benzi ($p>=1$)
3. Activitățile au și termen limită, profit...

Teme laborator Greedy

Problema rucsacului

- **Varianta continuă**:

Se consideră un rucsac de capacitate (greutate) maximă G și n obiecte caracterizate prin următoarele **numere reale pozitive**:

- greutățile lor $g_1,\ldots,g_n$ – numere **reale pozitive**
- câștigurile $c_1,\ldots,c_n$ obținute la încărcarea lor **<u>în totalitate</u>** în rucsac.

Problema rucsacului

- **Varianta continuă**:

Se consideră un rucsac de capacitate (greutate) maximă G și n obiecte caracterizate prin următoarele **numere reale pozitive**:

  - greutățile lor $g_1,\ldots,g_n$ – numere **reale pozitive**
  - câștigurile $c_1,\ldots,c_n$ obținute la încărcarea lor **<u>în totalitate</u>** în rucsac.

Din fiecare obiect poate fi încărcată orice fracțiune a sa.

Să se determine o modalitate de încărcare de (fracțiuni de) obiecte în rucsac, astfel încât **câștigul total** să fie maxim.

Problema rucsacului

- Exemplu

G	=	8			
n	=	4			
c	=	8	12	6	10
g	=	4	8	3	10
		Ob.1	Ob.2	Ob.3	Ob.4

Problema rucsacului

- **Variantă de enunț:**

Se dau

T=timp total de funcționare a unei resurse

n activități cu **durata** d_i și **profitul** p_i (dacă activitatea se execută în totalitate) care necesită resursa

O activitate începută poate fi întreruptă obținându-se un profit parțial.

Se cere: profitul maxim care se poate obține

Problema rucsacului

- Reprezentarea soluției

- Câștigul total corespunzător unei soluții posibile (funcția de optimizat)

Problema rucsacului

Care este primul obiect pe care îl încărcăm în rucsac + ce fracțiune din el?

cel cu câștigul pe unitate cel mai mare

Problema rucsacului

- Exemplu

$G = 8$

$n = 4$

	Ob.1	Ob.2	Ob.3	Ob.4
c =	8	12	6	10
g =	4	8	3	10
r =	2	1,5	2	1

r = câștigul pe unitatea de greutate

Soluția:

x = (1, 1/8, 1, 0)

Problema rucsacului

Pseudocod

- Presupunem $g_1+...+g_n>G$
- ordonăm obiectele descrescător după câștigul pe unitatea de greutate – presupunem obiectele renotate astfel încât

$$\frac{c_1}{g_1} \geq \frac{c_2}{g_2} \geq \ldots \geq \frac{c_n}{g_n}$$

Problema rucsacului

Pseudocod

```
G1 ← G { G1 = greutatea ramasa }
for i=1,n
   if gi ≤ G1
       xi←1; G1 ← G1-gi
   else
```

Problema rucsacului

Pseudocod

```
G1 ← G { G1 = greutatea ramasa }
for i=1,n
   if g_i ≤ G1
        x_i←1; G1 ← G1-g_i
   else
        x_i←G1/g_i;
        for j=i+1,n
            x_j ← 0
        stop
write(x) { x=(1,...,1,x_j,0,...,0) cu x_j∈[0,1).}
```

Problema rucsacului

Pseudocod

G1 $\leftarrow$ G { **G1 = greutatea ramasa** }

```
for i=1,n
    if gi ≤ G1
        xi←1; G1 ← G1-gi
    else
        xi←G1/gi;
        for j=i+1,n
            xj ← 0
        stop
```

write(x) { **x=(1,...,1,x_j,0,...,0)** cu $x_j \in [0,1)$.}

Complexitate O(nlogn)

Problema rucsacului

- **Este corect algoritmul?**

Problema rucsacului

- Este corect algoritmul?
- Dar dacă obiectele nu se pot fracționa?