Metoda GREEDY

Metoda Greedy

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

  Se dă o mulțime finită A.

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

  +

  îndeplinește un **criteriu de optim** (este **soluție optimă**), adică minimizează/maximizează o funcție obiectiv f

Metoda Greedy

- **Cadru formal**:
  - O mulțime finită $A=\{a_1,\ldots,a_n\}$
  - O funcție $f:\mathcal{P}(A)\to\mathbf{R}$ – **trebuie minimizată/maximizată**
  - O **proprietate** definită pe mulțimea submulțimilor lui A

    propr:$\mathcal{P}(A) \to \{0,1\}$ => **criteriul pentru ca un element să poată fi adăugat la soluția construită până la pasul curent**

O submulțime $S \subseteq A$ sn **soluție (posibilă)** dacă propr(S)=1

Să se determine o soluție posibilă care optimizează funcția f

Metoda Greedy

- **Probleme de optim**

- **Exemplu:** Dată o mulțime de intervale, să se determine o submulțime de cardinal maxim de intervale care nu se suprapun (activități cu intervale de desfășurare compatibile)

  - A = mulțimea de intervale
  - Soluție posibilă = o submulțime de intervale disjuncte

    propr(S) = True ⇔ S conține intervale disjuncte

  - Soluție optimă = soluție posibilă de **cardinal maxim**

Funcția obiectiv f(S) = |S|

Metoda Greedy

- **Strategie**: Se încearcă o construire directă a **unei** soluții optime, **element cu element**

- 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

- **Strategie**: 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 – v. pb monede/bancnote seminar**

Metoda Greedy

- **<u>Pb monede (seminar):</u>** Se dă o sumă S și avem la dispoziție monede cu valorile: 1, 5, 10, 25 (un număr nelimitat de monede). Să se determine o modalitate de a plăti suma S folosind un număr minim de monede.

  Algoritmul propus este corect și dacă aveam bancnote cu valorile 1, 10, 30, 40 (de exemplu S=60) ?

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

Metoda Greedy

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

  - **Varianta 1** – **fără prelucrare inițială**:

```
S←ϕ
for i=1,n
    x ← alege(A)
    A ← A - {x}
    if propr(S∪{x})=1
        S ← S ∪ {x}
return S
```

Metoda Greedy

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

  - **Varianta 2** – **cu prelucrare inițială:**

```
prelucreaza(A)
S←ϕ
for x in A
    if propr(S ∪ x)=1
        S ← S ∪ {x}
return S
```

Metoda Greedy

- În algoritm nu apare funcţia f $\Rightarrow$

  nevoie de a demonstra corectitudinea

Metoda Greedy

- <u>**Exemplul 1**</u> Se consideră mulțimea de valori reale $A=\{a_1,...,a_n\}$. Să se determine o submulțime a lui A a cărei sumă a elementelor este maximă.

Metoda Greedy

- <u>**Exemplul 1**</u> Se consideră mulțimea de valori reale $A=\{a_1,...,a_n\}$. Să se determine o submulțime a lui A a cărei sumă a elementelor este maximă.

- Soluție S = mulțimea elementelor pozitive, dacă există,
  sau cel mai mare element din mulțime, altfel

Metoda Greedy

- **Exemplul 2** Se consideră mulțimea de valori întregi $A=\{a_1,\ldots,a_n\}$ și $k<n$. Să se determine o submulțime cu k elemente a lui A cu suma a elementelor este maximă.

Metoda Greedy

- **<u>Exemplul 2</u>** Se consideră mulțimea de valori întregi $A=\{a_1,...,a_n\}$ și $k<n$. Să se determine o submulțime cu k elemente a lui A cu suma a elementelor este maximă.

- Soluție S = mulțimea celor mai mari k elemente din A

Metoda Greedy

- **<u>Exemplul 2</u>** Se consideră mulțimea de valori întregi $A=\{a_1,\ldots,a_n\}$ și $k<n$. Să se determine o submulțime cu k elemente a lui A cu suma a elementelor este maximă.

- Soluție S = mulțimea celor mai mari k elemente din A

Fără prelucrare inițială	Cu ordonare inițială

S←φ
for i=1,n
    x ← maxim(A)
    A ← A - {x}
    S ← S ∪ {x}
    if |S| = k stop
	S←φ
	sort_descrescator(A)
	for x in A
	    S ← S ∪ {x}
	    if |S| = k stop

Complexitate	Complexitate

Metoda Greedy

- **Exemplul 2** Se consideră mulțimea de valori reale $A=\{a_1,\ldots,a_n\}$ și $k<n$. Să se determine o submulțime cu k elemente a lui A cu suma a elementelor este maximă.

- Soluție S = mulțimea celor mai mari k elemente din A

Fără prelucrare inițială	Cu ordonare inițială

`S←ϕ`
`for i=1,n`
`    x ← maxim(A)`
`    A ← A - {x}`
`    S ← S ∪ {x}`
`    if |S| = k  stop`
	`S←ϕ`
	`sort_descrescator(A)`
	`for x in A`
	`    S ← S ∪ {x}`
	`    if |S| = k  stop`

Complexitate **O(nk)**	Complexitate **O(nlog(n))**

Metoda Greedy

- **<u>Exemplul 3</u> (Contraexemplu)** Se consideră mulțimea $A=\{a_1,...,a_n\}$ cu elemente **pozitive și un număr natural M**. Să se determine o submulțime a lui A de sumă maximă, dar cel mult egală cu o valoare M dată.

Exemple

Probleme de planificare

- Se dau n activități care trebuie să folosească o resursă comună. Doar o activitatea poate folosi resursa la un moment dat și o activitate începută nu poate fi întreruptă

Probleme de planificare

- Se dau n activități care trebuie să folosească o resursă comună. Doar o activitatea poate folosi resursa la un moment dat și o activitate începută nu poate fi întreruptă

A. **Planificarea activităților astfel încât să fie minimizat timpul mediu de așteptare:** Știind **duratele** activităților, să se determine ordinea în care se execută activitățile astfel încât timpul mediu de așteptare să fie minim (=**media timpilor necesari finalizării tuturor activităților**: o activitate trebuie să aștepte se finalizeze activitățile programate înainte pentru a putea începe)

Probleme de planificare

- Se dau n activități care trebuie să folosească o resursă comună. Doar o activitatea poate folosi resursa la un moment dat și o activitate începută nu poate fi întreruptă

B. **Planificarea activităților fără suprapuneri**: Știind **intervalele de desfășurare** ale activităților, să se determine un număr maxim de activități compatibile (cu intervale de desfășurare disjuncte)

Probleme de planificare

- Se dau n activități care trebuie să folosească o resursă comună. Doar o activitatea poate folosi resursa la un moment dat și o activitate începută nu poate fi întreruptă

C. **Determinarea numărului minim de resurse necesare pentru a putea efectua toate activitățile,** date fiind **intervalele de desfășurare** ale activităților (o resursă poate fi folosită doar de activități compatibile = cu intervale de desfășurare disjuncte)