Structuri de date și algoritmi

Tehnica Greedy

Lect. Dr. Ing. Cristian Chilipirea – cristian.chilipirea@mta.ro

Greedy

"A greedy algorithm always makes the choice that looks best at the moment. That is, it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution."

– Cormen, Introduction To Algorithms

Greedy

Când ai mai multe opțiuni alege pe cea mai bună.

Ce înseamnă cea mai bună?
După ce metrică?

Greedy

Când ai mai multe opțiuni alege pe cea mai bună.

Ce înseamnă cea mai bună?

După ce metrică?

Depinde de problemă

Problemă: Minimizare timp servire

- Timp servire = Timp așteptare + Timp execuție
- Toți clienții vin simultan. Vrem să îi preluăm într-o ordine în așa fel încât să minimizăm timpul de servire total.

Problemă: Minimizare timp servire

- Timp servire = Timp așteptare + Timp execuție
- Toți clienții vin simultan. Vrem să îi preluăm într-o ordine în așa fel încât să minimizăm timpul de servire total.
- Exemplu: Fie task-uri care durează **5**, **10** respectiv **4**.

Sarcina	Timp în Sistem
1	**5**(execuţie)=5
2	5(aşteptare)+**10**(execuţie)=15
3	5(aşteptare)+10(aşteptare)+**4**(execuţie)=19

Timp total 5+15+19

Problemă: Minimizare timp servire

Exemplu: Fie task-uri care durează $t_1 = 5$, $t_2 = 10$ respectiv $t_3 = 4$

Ordine	Timp Total
(1,2,3)	5+(5+10)+(5+10+4)=39
(1,3,2)	5+(5+4)+(5+4+10)=33
(2,1,3)	10+(10+5)+(10+5+4)=44
(2,3,1)	10+(10+4)+(10+4+5)=43
(3,1,2)	**4+(4+5)+(4+5+10)=32**
(3,2,1)	4+(4+10)+(4+10+5)=37

Problemă: Minimizare timp servire

Exemplu: Fie task-uri care durează $t_1 = 5$, $t_2 = 10$ respectiv $t_3 = 4$

Ordine	Timp Total
(1,2,3)	5+(5+10)+(5+10+4)=39
(1,3,2)	5+(5+4)+(5+4+10)=33
(2,1,3)	10+(10+5)+(10+5+4)=44
(2,3,1)	10+(10+4)+(10+4+5)=43
(3,1,2)	**4+(4+5)+(4+5+10)=32**
(3,2,1)	4+(4+10)+(4+10+5)=37

Alegem mereu timpul de execuție cel mai scurt.

Problemă: Minimizare timp servire – Demonstrație

- Fie timpii $T = \{t_1, t_2, t_2, \dots, t_n\}$
- Fie permutarea $T_\sigma = \{t_{\sigma 1}, t_{\sigma 2}, t_{\sigma 2}, \dots, t_{\sigma n}\}$

$$T_{total} = n * t_{\sigma 1} + (n-1) * t_{\sigma 2} + \ \dots + 1 * t_{\sigma n}$$

T_{total} este minim dacă $\forall i, j;\ i < j$ atunci $t_{\sigma i} \leq t_{\sigma j}$

Problemă: Număr minim de monede

- Se dau monede de valori diferite (1, 5, 10 ron)
- Se dorește numărul minim de monede pentru a ajunge la o sumă dată (sau aproape)

Problemă: Număr minim de monede

- Se dau monede de valori diferite (1, 5, 10 ron)
- Se dorește numărul minim de monede pentru a ajunge la o sumă dată (sau aproape)
- Fie suma: 26. Ce monede alegem?

Problemă: Număr minim de monede

- Se dau monede de valori diferite (1, 5, 10 ron)
- Se dorește numărul minim de monede pentru a ajunge la o sumă dată (sau aproape)
- Fie suma: 26. Ce monede alegem?

10

Problemă: Număr minim de monede

- Se dau monede de valori diferite (1, 5, 10 ron)
- Se dorește numărul minim de monede pentru a ajunge la o sumă dată (sau aproape)
- Fie suma: 26. Ce monede alegem?

10, 10

Problemă: Număr minim de monede

- Se dau monede de valori diferite (1, 5, 10 ron)
- Se dorește numărul minim de monede pentru a ajunge la o sumă dată (sau aproape)
- Fie suma: 26. Ce monede alegem?

10, 10, 5

Problemă: Număr minim de monede

- Se dau monede de valori diferite (1, 5, 10 ron)
- Se dorește numărul minim de monede pentru a ajunge la o sumă dată (sau aproape)
- Fie suma: 26. Ce monede alegem?

10, 10, 5, 1

Problemă: Număr minim de monede

- Se dau monede de valori diferite (1, 5, 10 ron)
- Se dorește numărul minim de monede pentru a ajunge la o sumă dată (sau aproape)
- Fie suma: 26. Ce monede alegem?

10, 10, 5, 1

La fiecare pas alegem cea mai **mare** monedă mai mică decât suma rămasă.

Problemă: Număr minim de monede

- Monede de: 1, 5, 12, 18, 25
- Suma: 36

La fiecare pas alegem cea mai **mare** monedă mai mică decât suma rămasă.

25

Sumă rămasă: 36 – 25 = 11

Problemă: Număr minim de monede

- Monede de: 1, 5, 12, 18, 25
- Suma: 36

La fiecare pas alegem cea mai **mare** monedă mai mică decât suma rămasă.

25 + 5

Sumă rămasă: 36 – 25 – 5 = 6

Problemă: Număr minim de monede

- Monede de: 1, 5, 12, 18, 25
- Suma: 36

La fiecare pas alegem cea mai **mare** monedă mai mică decât suma rămasă.

25 + 5 + 5

Sumă rămasă: 36 – 25 – 5 – 5 = 1

Problemă: Număr minim de monede

- Monede de: 1, 5, 12, 18, 25
- Suma: 36

La fiecare pas alegem cea mai **mare** monedă mai mică decât suma rămasă.

25 + 5 + 5 + 1

Sumă rămasă: 36 – 25 – 5 – 5 – 1 = 0

Problemă: Număr minim de monede

- Monede de: 1, 5, 12, 18, 25
- Suma: 36

La fiecare pas alegem cea mai **mare** monedă mai mică decât suma rămasă.

25 + 5 + 5 + 1

Sumă rămasă: 36 – 25 – 5 – 5 – 1 = 0

Dar se poate din mai puține monede?

Problemă: Număr minim de monede

- Monede de: 1, 5, 12, 18, 25
- Suma: 36

La fiecare pas alegem cea mai **mare** monedă mai mică decât suma rămasă.

25 + 5 + 5 + 1

Sumă rămasă: 36 – 25 – 5 – 5 – 1 = 0

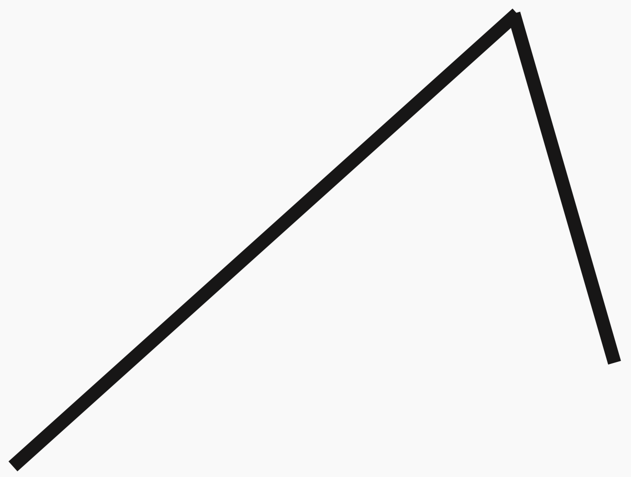
Dar se poate din mai puține monede?

12 + 12 + 12

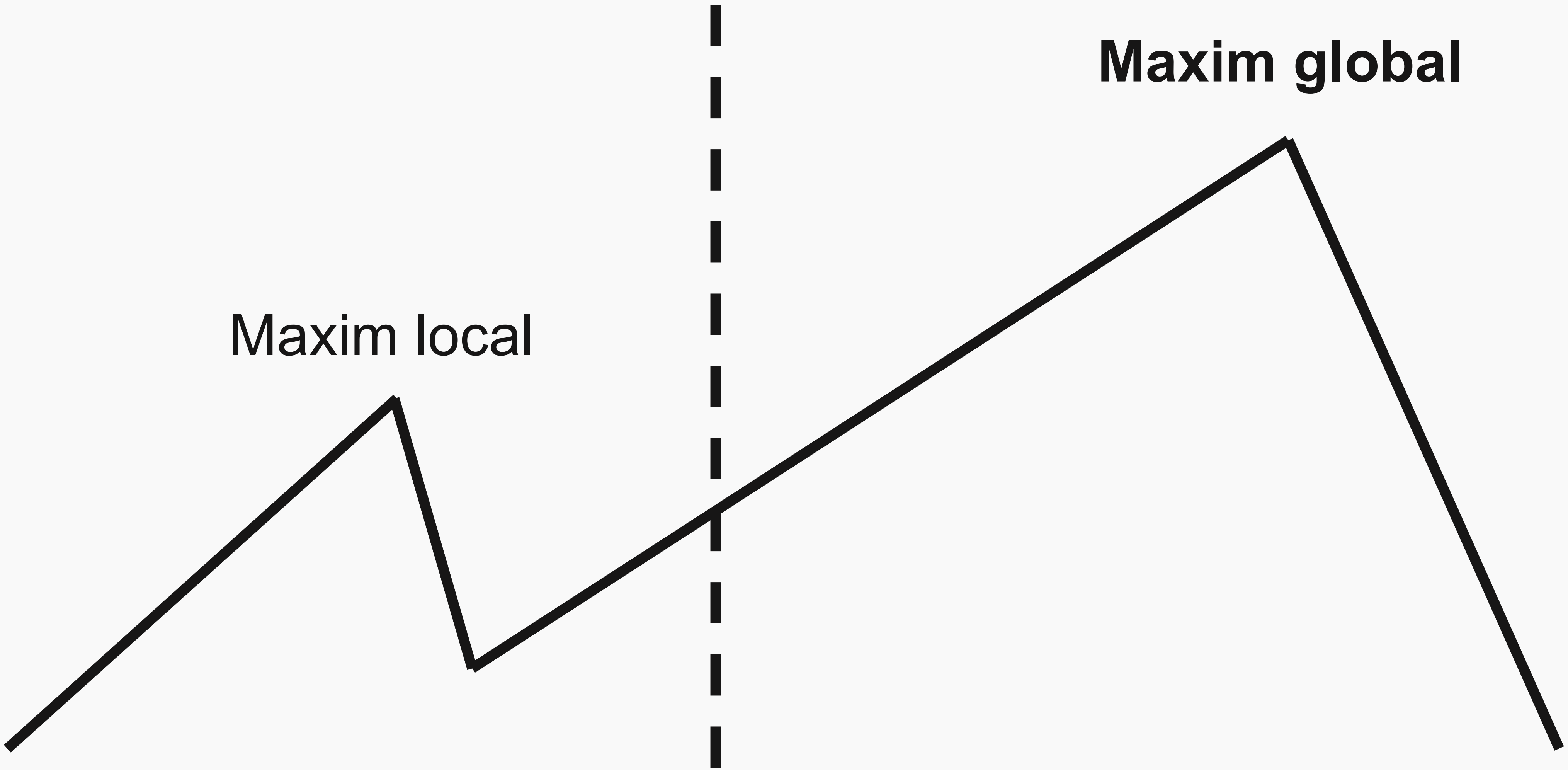
18 + 18

Soluție locală vs soluție globală

Găsirea celui mai înalt punct dintr-un munte folosind greedy:
Dacă o poziție de lângă noi e mai sus de unde ne aflăm, ne urcăm pe aceasta.

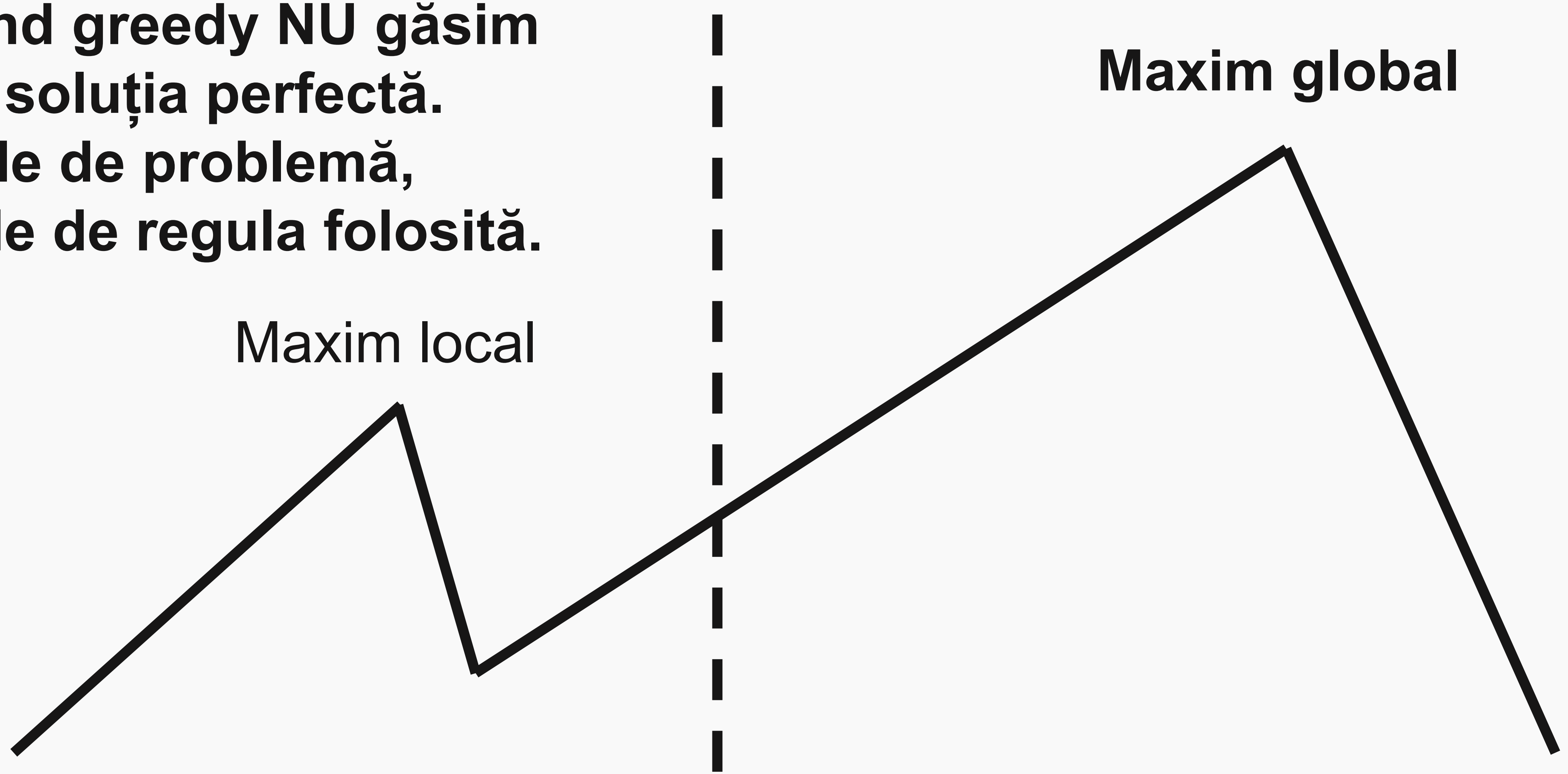


Soluție locală vs soluție globală

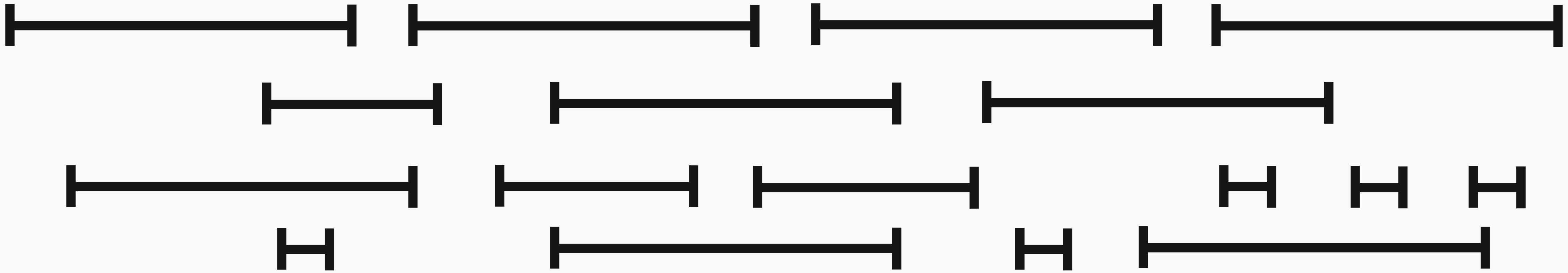


Soluție locală vs soluție globală

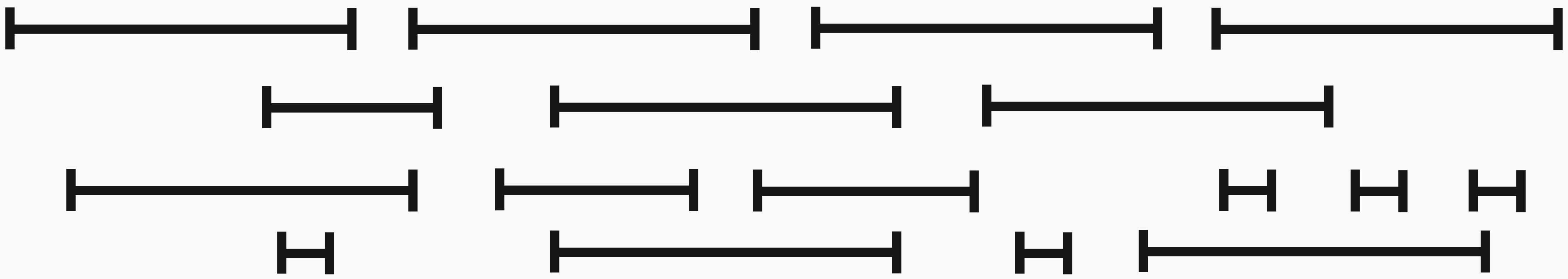
Folosind greedy NU găsim mereu soluția perfectă. Depinde de problemă, depinde de regula folosită.



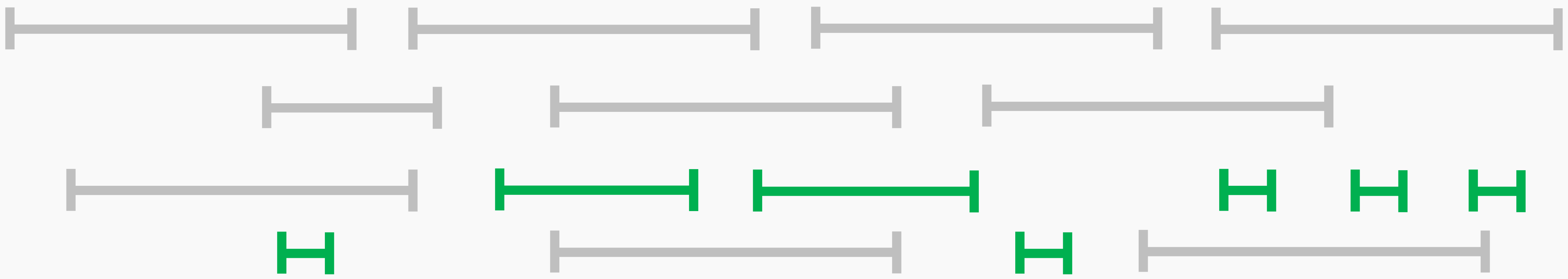
Problemă: Număr maxim intervale disjuncte



Problemă: Activity Selection



Problemă: Număr maxim intervale disjuncte

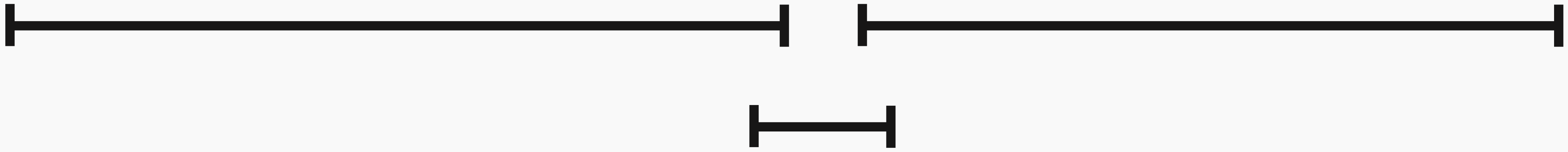


~~Soluție~~ greedy 1

Alegem intervalele cele mai scurte

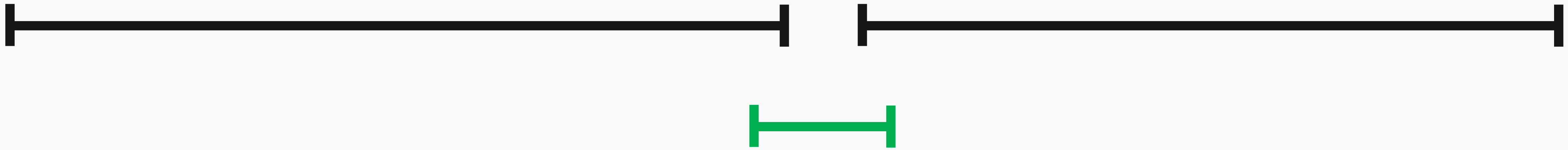
~~Soluție~~ greedy 1

Alegem intervalele cele mai scurte



~~Soluție~~ greedy 1

Alegem intervalele cele mai scurte

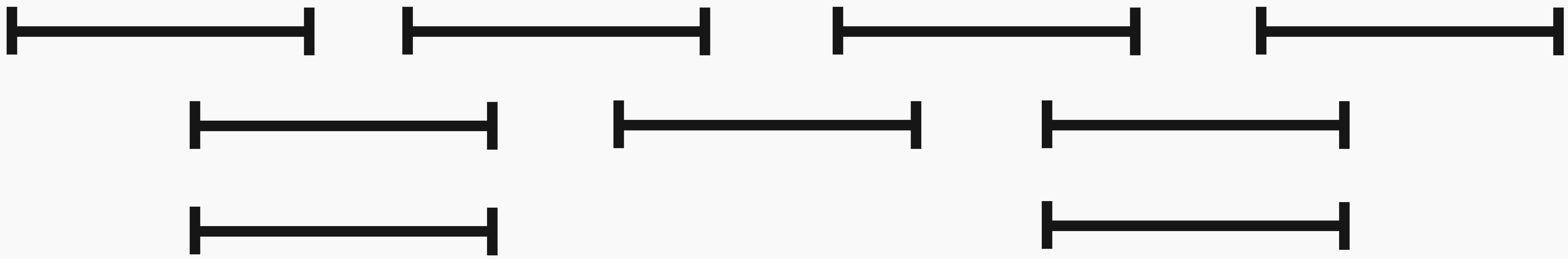


~~Soluție~~ greedy 2

Alegem intervalele cu cele mai puține suprapuneri

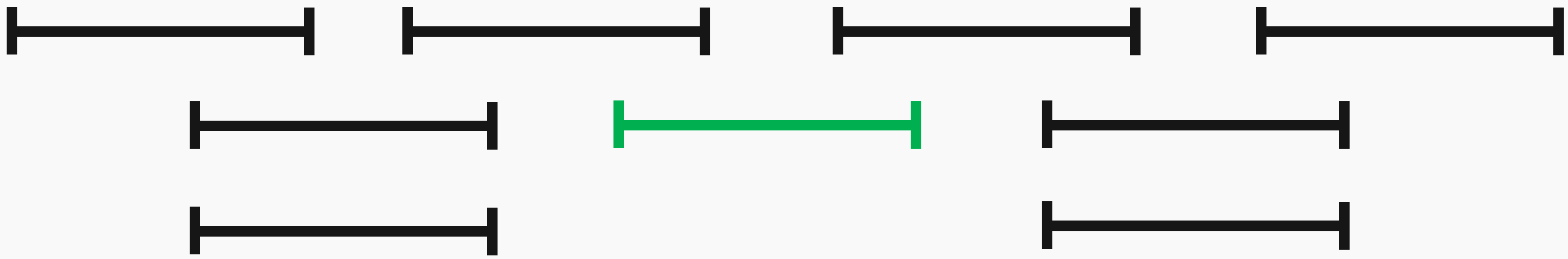
~~Soluție~~ greedy 2

Alegem intervalele cu cele mai puține suprapuneri



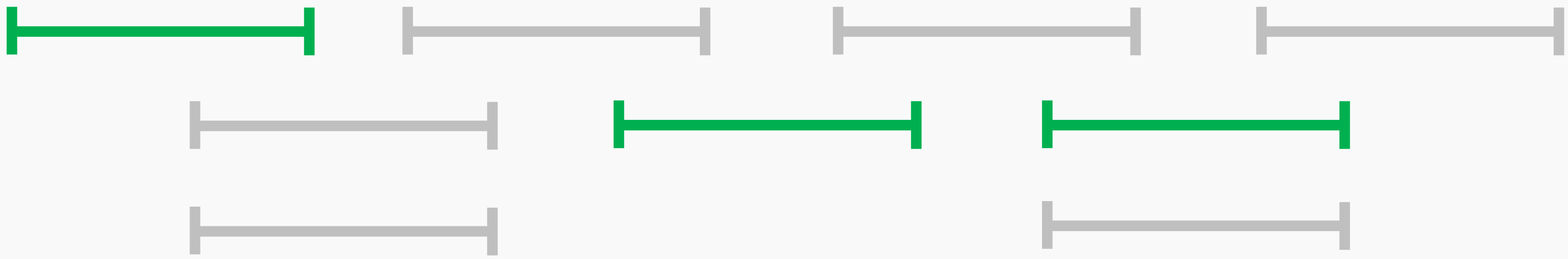
~~Soluție~~ greedy 2

Alegem intervalele cu cele mai puține suprapuneri



~~Soluție~~ greedy 2

Alegem intervalele cu cele mai puține suprapuneri



~~Soluție~~ greedy 3

Alege intervale cu punctul de start cel mai mic

~~Soluție~~ greedy 3

Alege intervale cu punctul de start cel mai mic

~~Soluție~~ greedy 3

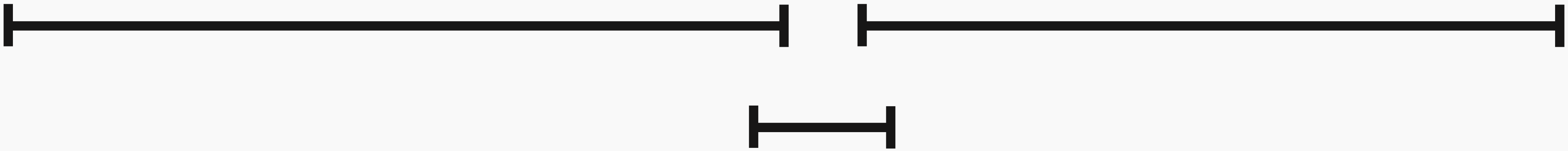
Alege intervale cu punctul de start cel mai mic

Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)

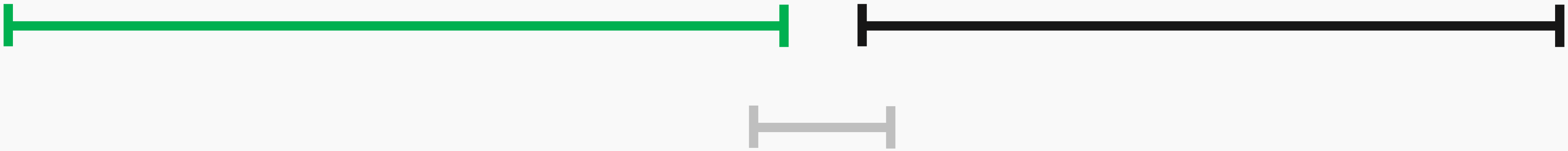
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



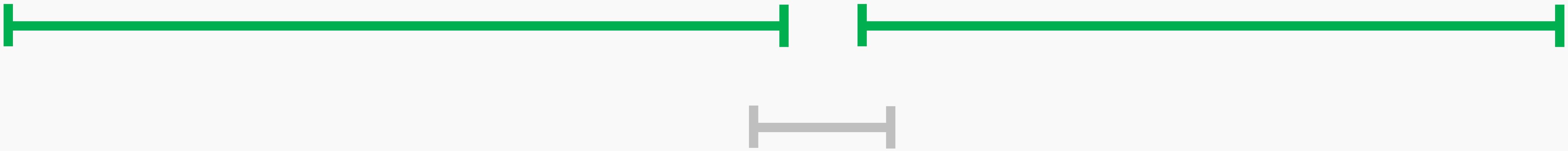
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



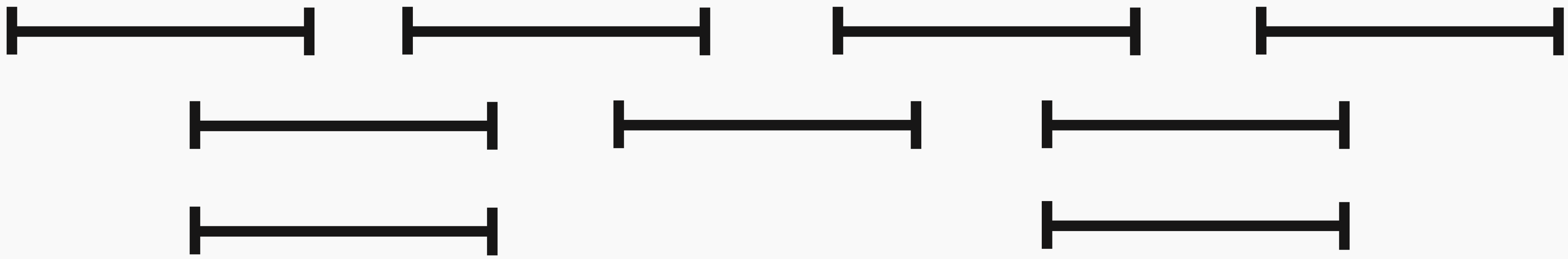
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



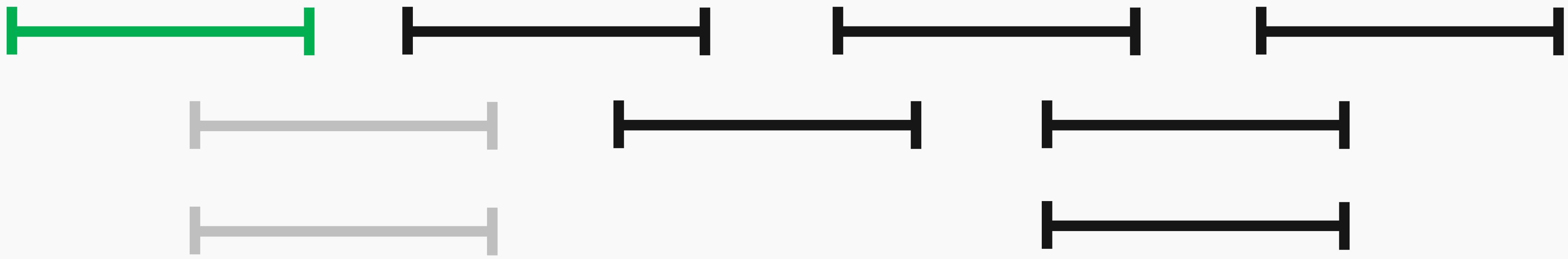
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



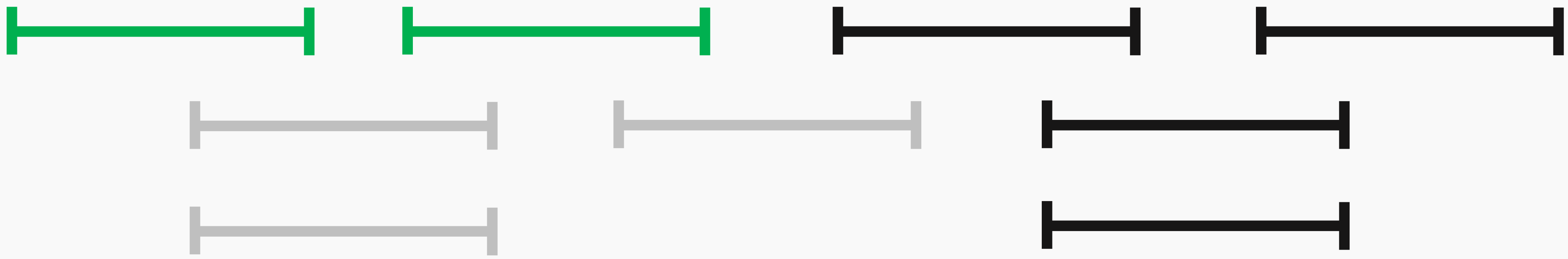
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



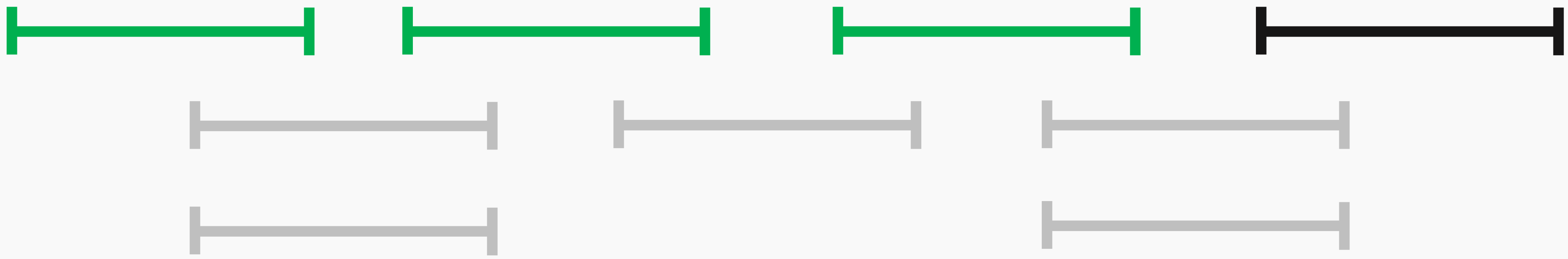
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



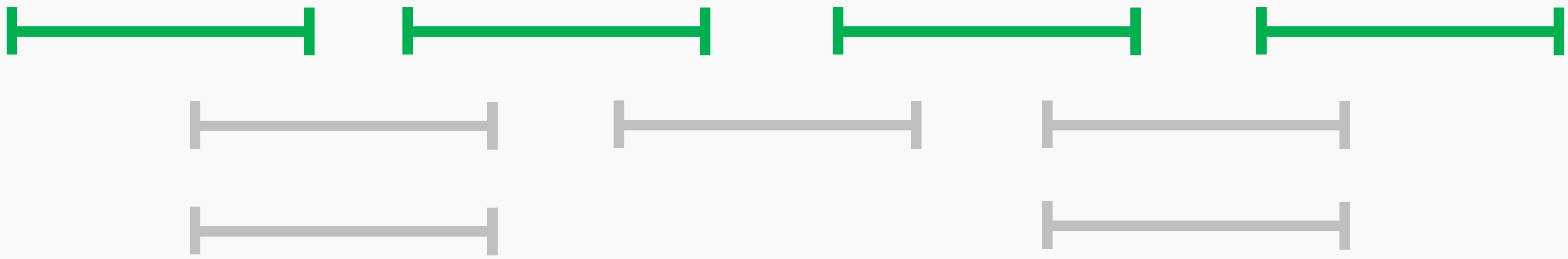
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)

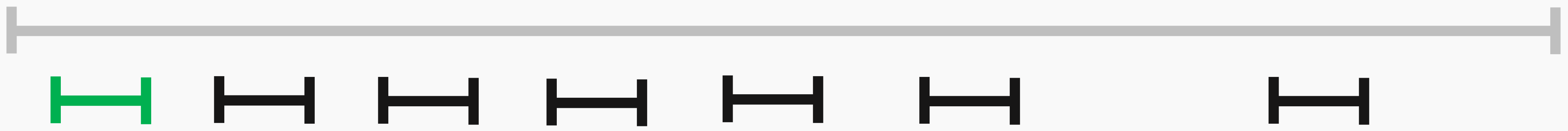


Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)

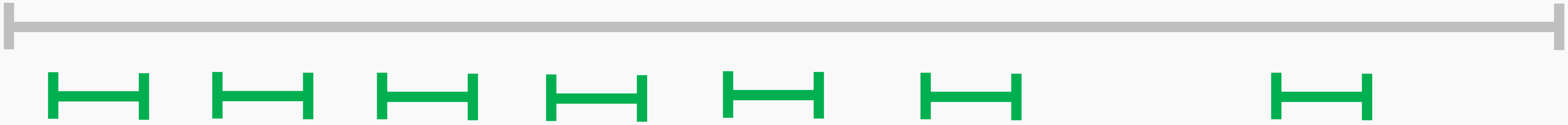
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



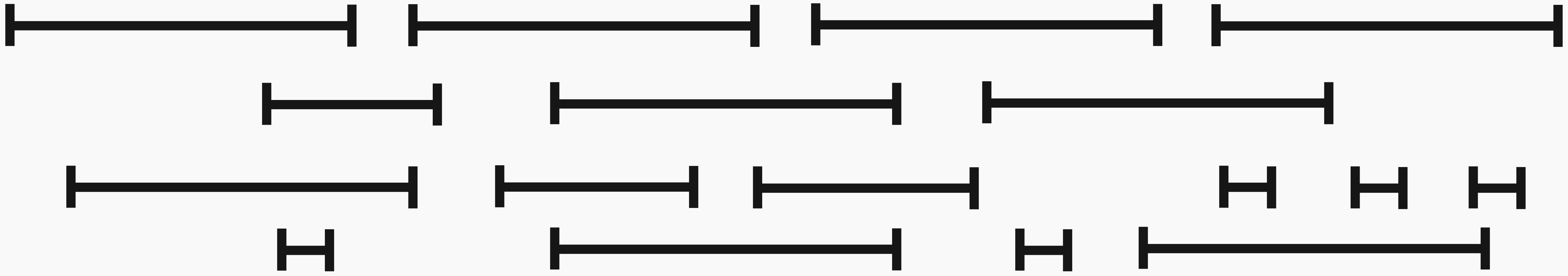
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



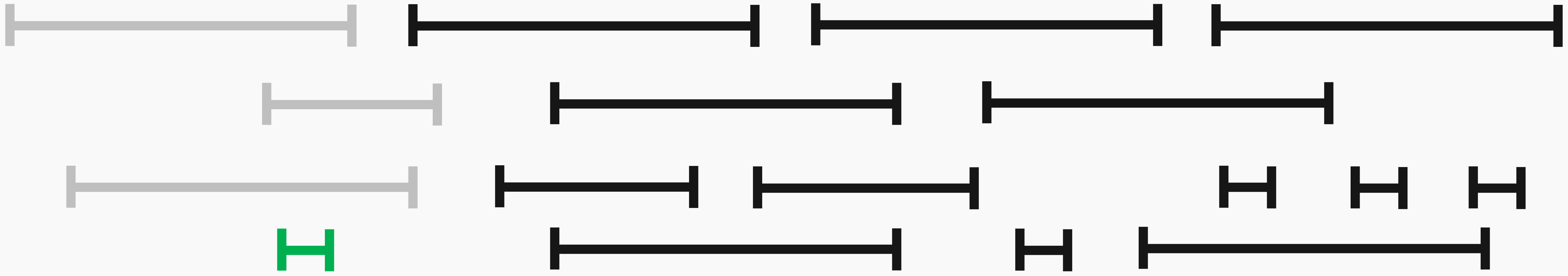
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



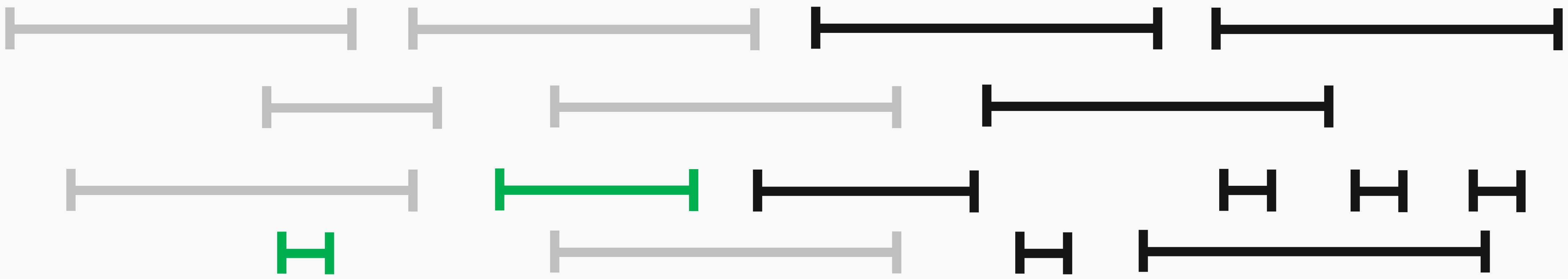
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



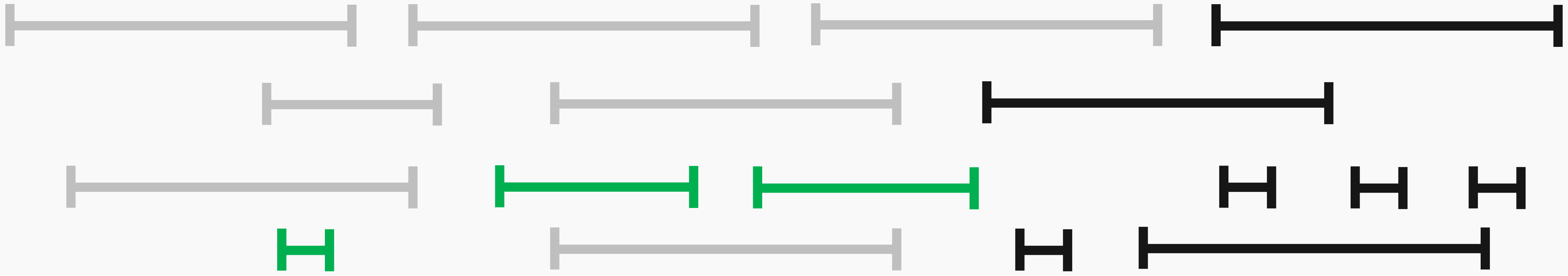
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



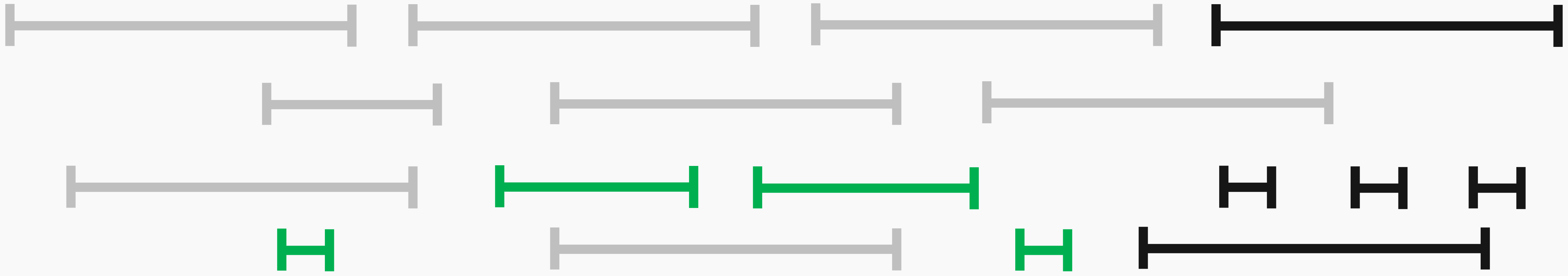
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



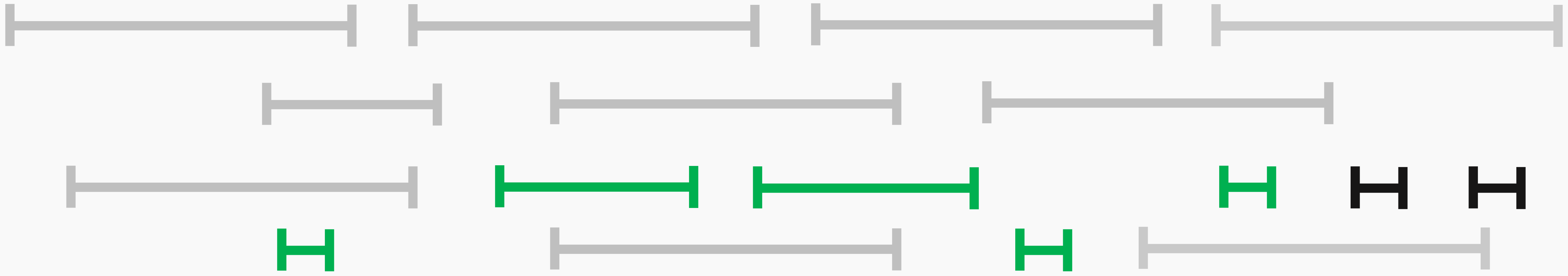
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



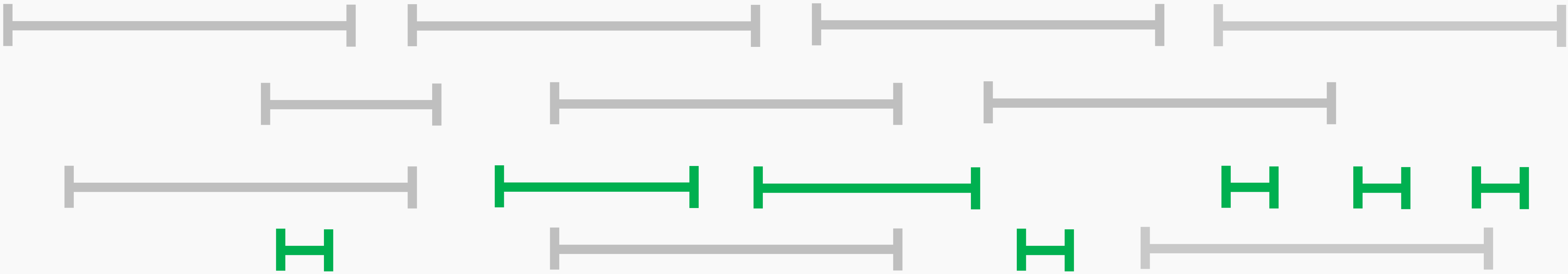
Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



Soluția greedy

Alegem intervalele cu cea mai mică valoare de oprire (dreapta)



Demonstrație corectitudine

Teoremă: Fie S_k un subset de intervale (subproblemă) și a_m un interval cu cea mai mică valoare de sfârșit ($a_m.s$) din S_k. Atunci a_m face parte din subsetul maxim de intervale disjuncte din S_k.

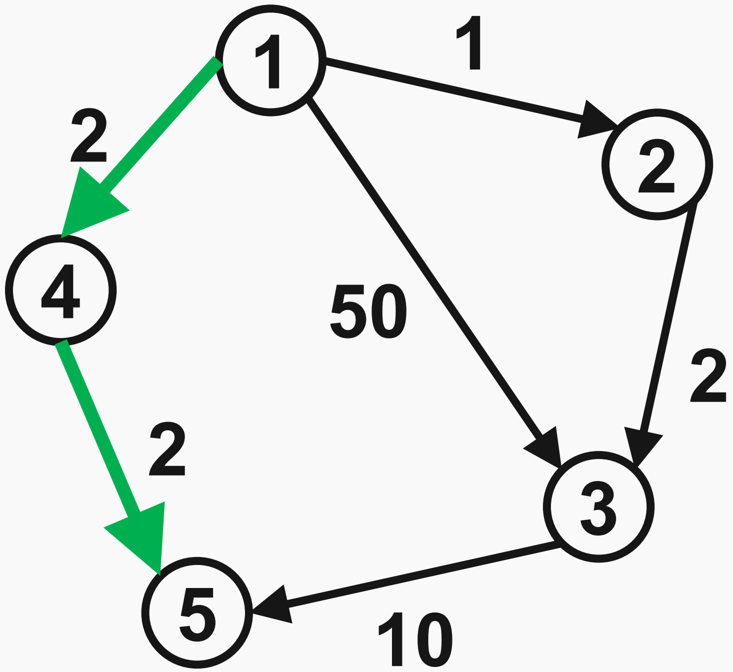
Demonstrație:

- Fie $\boldsymbol{A_k}$ cel mai mare subset de intervale disjuncte din S_k.
- Fie $\boldsymbol{a_j}$ are cea mai mică valoare de sfârșit ($\boldsymbol{a_j.s}$) a unui interval din $\boldsymbol{A_k}$
  - Dacă $\boldsymbol{a_j} = a_m$ am terminat
  - Dacă $\boldsymbol{a_j} \neq a_m$. Atunci fie $\boldsymbol{A'_k} = \boldsymbol{A_k} - \{\boldsymbol{a_j}\} \cup \{a_m\}$

  Intervalele din $\boldsymbol{A'_k}$ sunt disjuncte deoarece $\boldsymbol{a_j.s} \leq a_m.s$

  Deoarece $|\boldsymbol{A'_k}| = |\boldsymbol{A_k}|$ atunci $\boldsymbol{A'_k}$ include a_m și este subset maxim de intervale disjuncte din S_k.

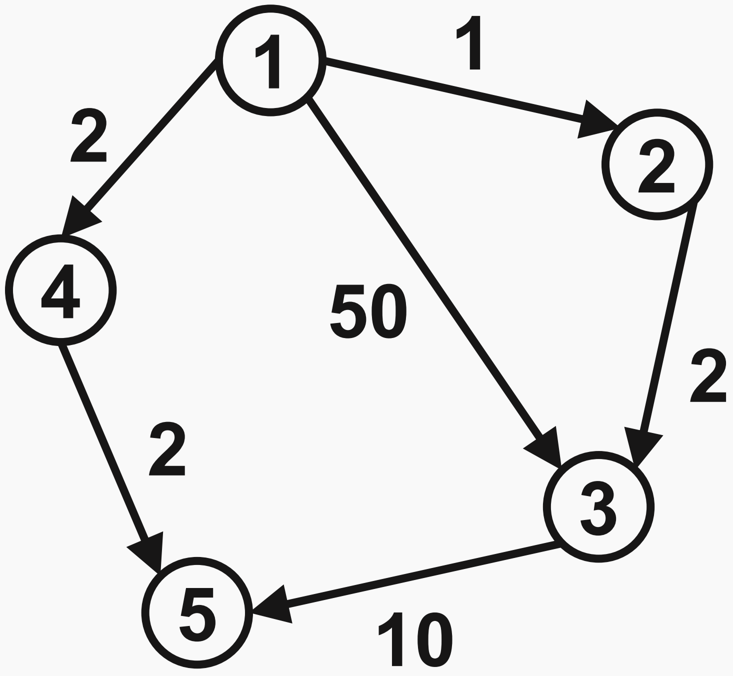
Drum minim în graf - greedy



Drum minim 1->5?

Drum minim în graf - greedy

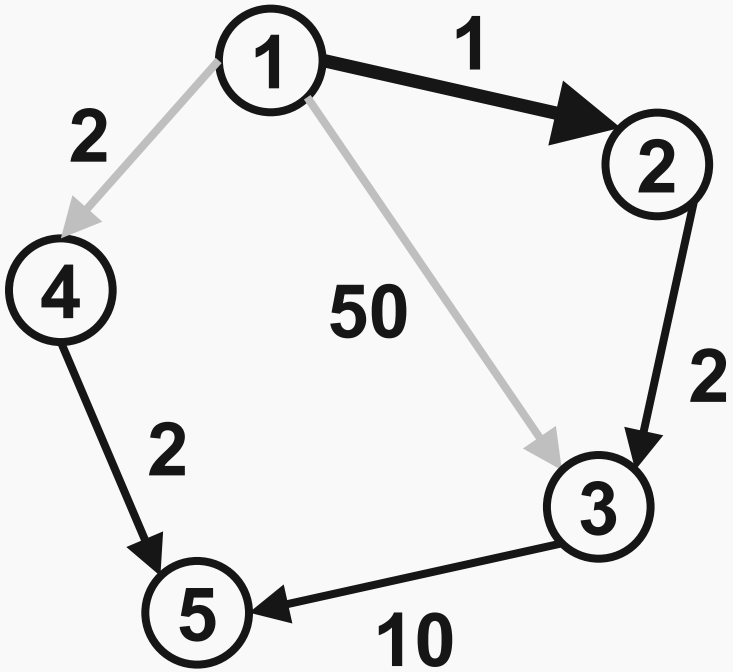
Alegem mereu muchia cu costul cel mai mic



Drum minim 1->5?

Drum minim în graf - greedy

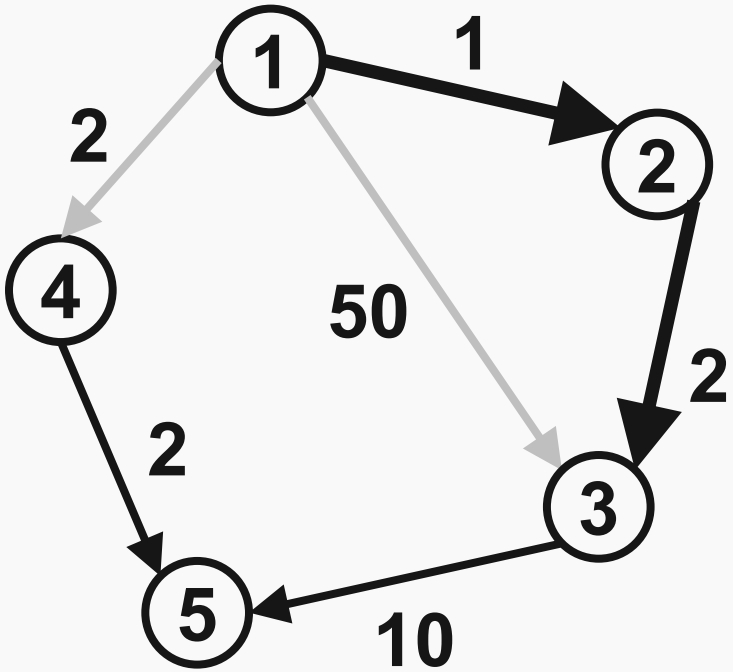
Alegem mereu muchia cu costul cel mai mic



Drum minim 1->5?

Drum minim în graf - greedy

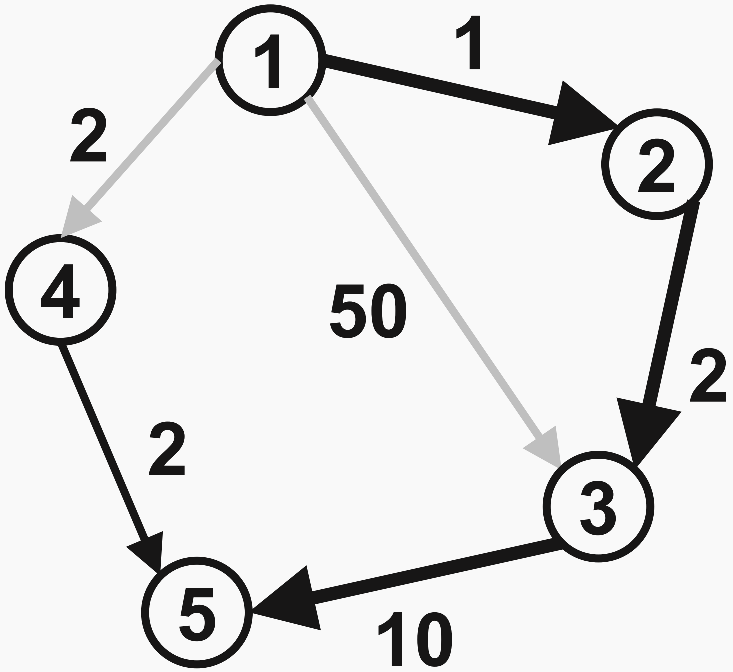
Alegem mereu muchia cu costul cel mai mic



Drum minim 1->5?

Drum minim în graf - greedy

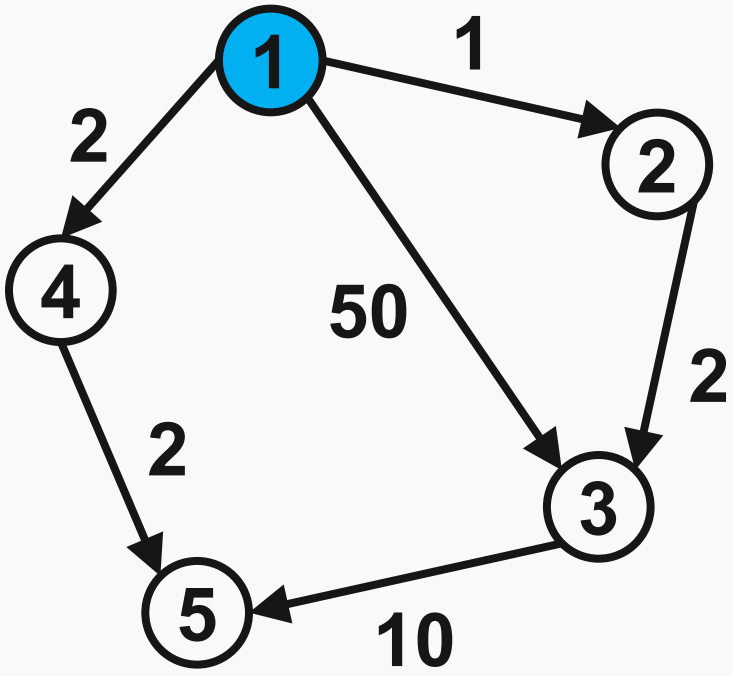
Alegem mereu muchia cu costul cel mai mic



Drum minim 1->5?

Drum minim în graf - greedy

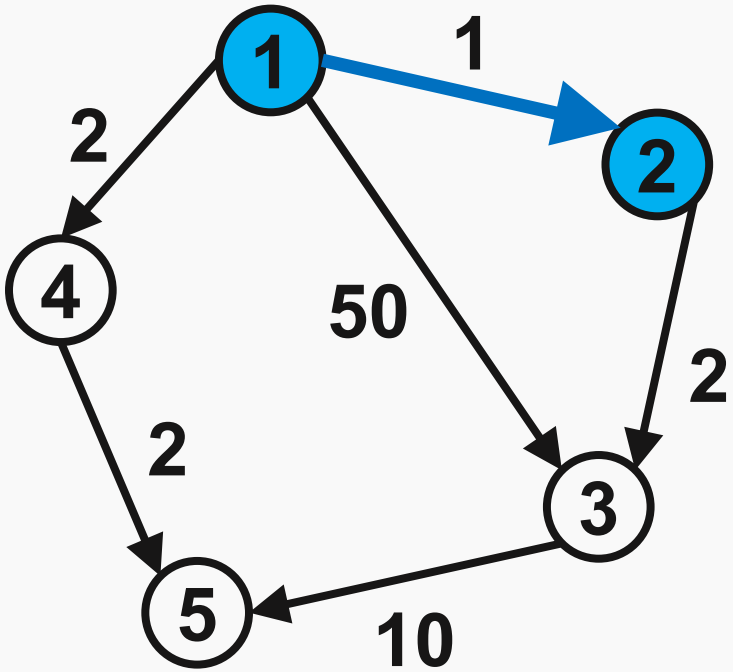
Alegem mereu muchia cu costul cel mai mic din toate "accesibile de zona explorată"



Drum minim 1->5?

Drum minim în graf - greedy

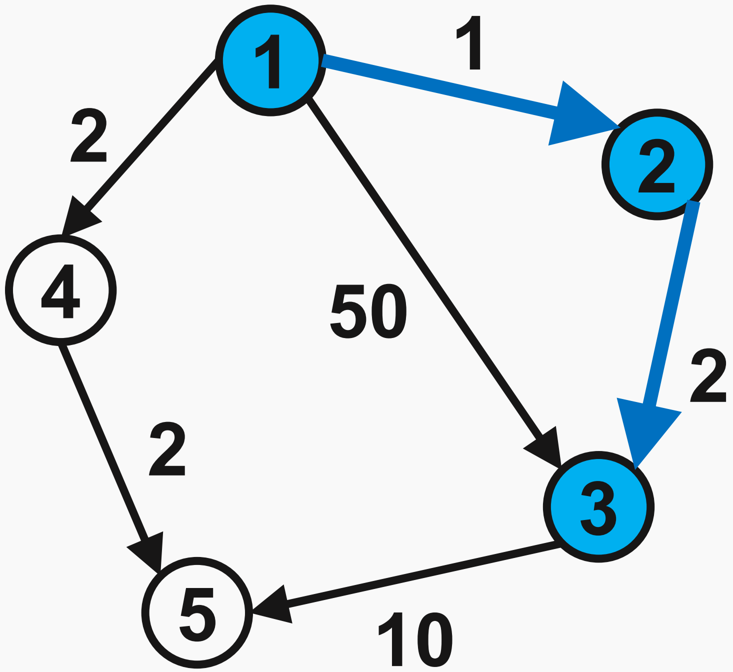
Alegem mereu muchia cu costul cel mai mic din toate "accesibile de zona explorată"



Drum minim 1->5?

Drum minim în graf - greedy

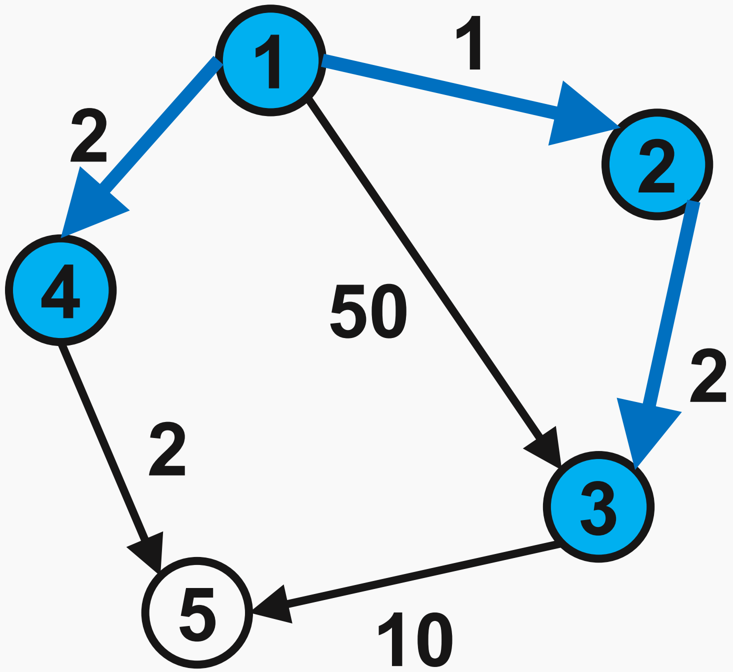
Alegem mereu muchia cu costul cel mai mic din toate "accesibile de zona explorată"



Drum minim 1->5?

Drum minim în graf - greedy

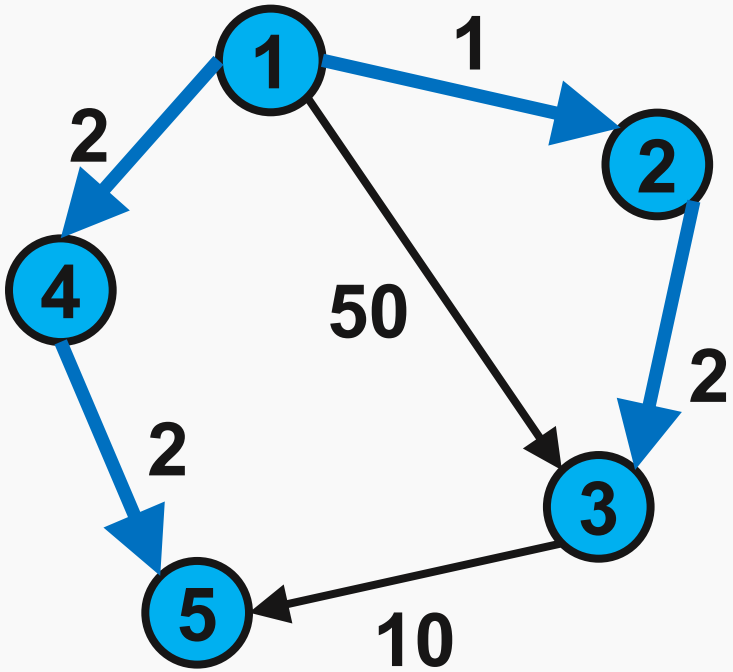
Alegem mereu muchia cu costul cel mai mic din toate "accesibile de zona explorată"



Drum minim 1->5?

Drum minim în graf - greedy

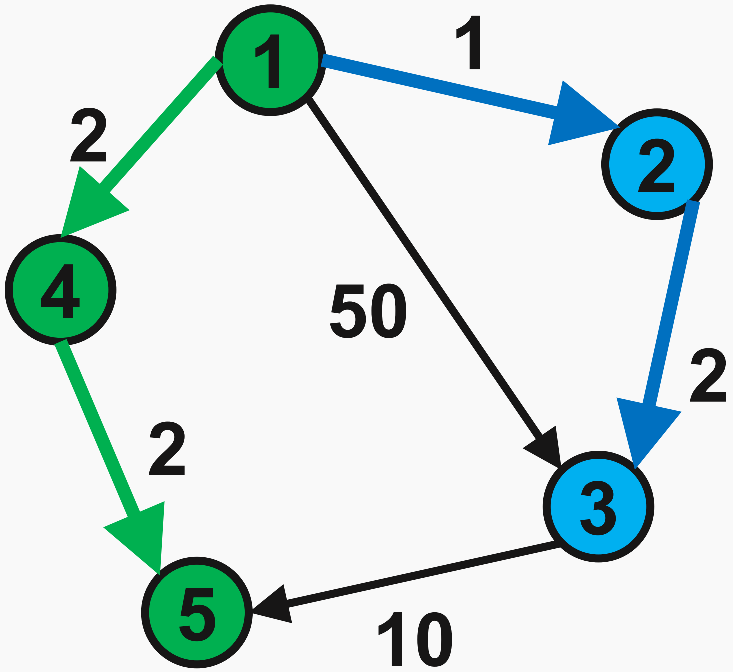
Alegem mereu muchia cu costul cel mai mic din toate "accesibile de zona explorată"



Drum minim 1->5?

Drum minim în graf - greedy

Alegem mereu muchia cu costul cel mai mic din toate "accesibile de zona explorată"

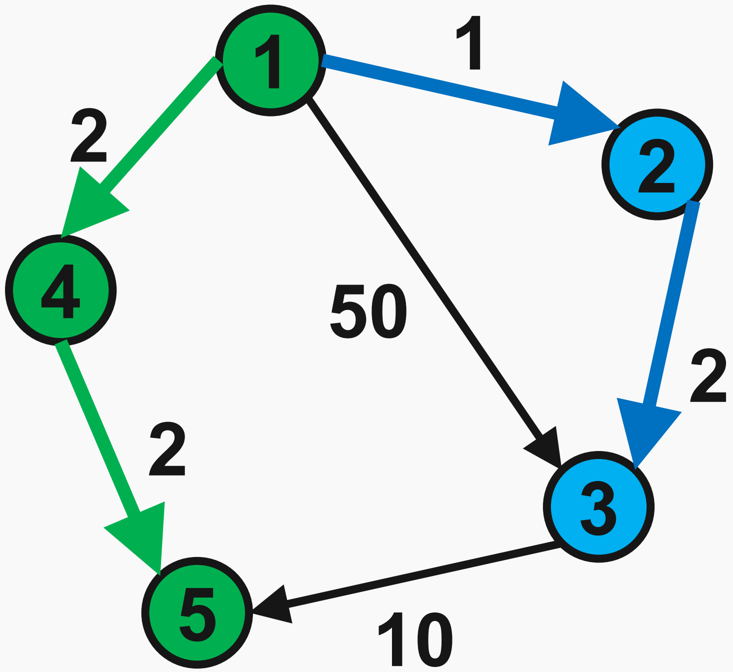


Drum minim 1->5?

Drum minim în graf - greedy

Alegem mereu muchia cu costul cel mai mic din toate "accesibile de zona explorată"

Ce algoritm este acesta?

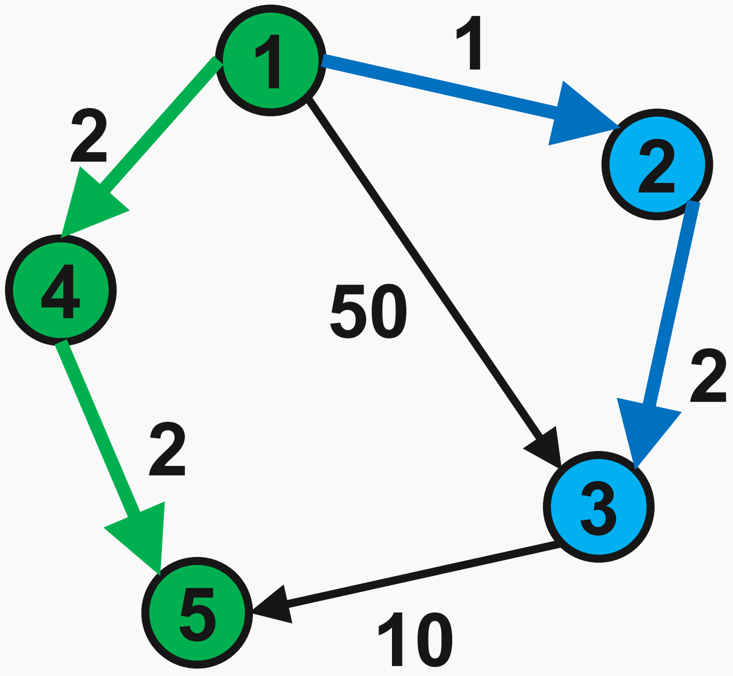


Drum minim 1->5?

Drum minim în graf - greedy

Alegem mereu muchia cu costul cel mai mic din toate "accesibile de zona explorată"

Ce algoritm este acesta?
Dijsktra



Drum minim 1->5?

Problema rucsacului

- Se dă un set de obiecte cu o greutate și un preț.
- Se dă un rucsac în care poate fi cărată o greutate maximă.
- Care obiecte să fie adăugate în rucsac pentru a maximiza valoarea?

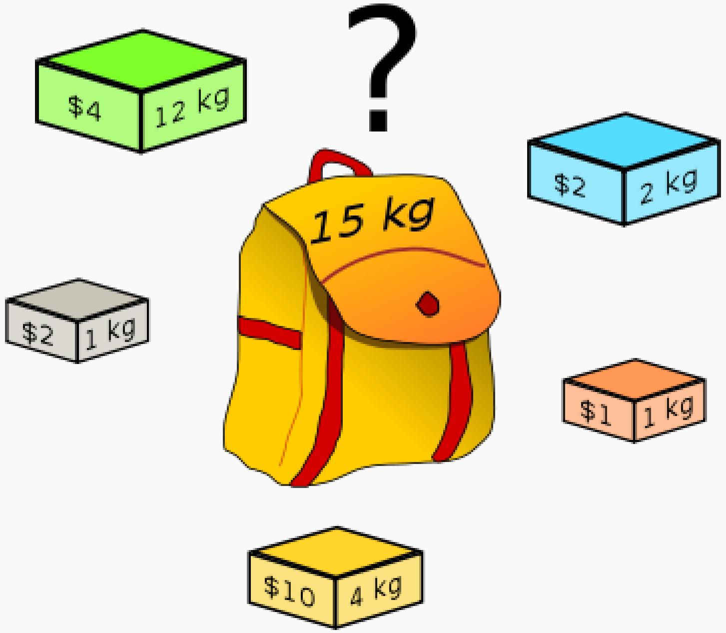


Image from: https://en.wikipedia.org/wiki/Knapsack_problem