NOTIȚE CURS 10 (amf.):

Metoda Greedy

3) Problema rucsacului (Knapsack Problem)

n obiecte → $g_1, g_2, \dots g_n$ greutățile obiectelor
$c_1, c_2, \dots c_n$ câștiguri integrale

G = capacitatea rucsacului

? O modalitate de încărcare a rucsacului a.î. câștigul total să fie maxim

Variante → continuă (orice obiect poate fi "tăiat") → Greedy → $O(n \log_2 n)$
→ discretă / var 0/1 (orice obiect poate fi încărcat doar integral

Varianta continuă

Câștig unitar $= \frac{c_i}{g_i}$ (câștig / greutate)

Algoritm Greedy $O(n \log_2 n)$

1) Sortăm obiectele descresc. după câștigul unitar.

2) Pt. fiecare obiect : dacă ob. încape complet în rucsac ⇒ îl încărcăm complet.
$O(n)$ dacă ob. nu încape complet ⇒ încărcăm o parte din el a.î. să umplem rucsacul. (și STOP)

Complexitate = $O(n \log_2 n)$

$G = 65$ Kg

$n = 7$

$\quad\quad O_1 \quad O_2 \quad O_3 \quad O_4 \quad O_5 \quad O_6 \quad O_7$

$c = (100, 50, 150, 100, 20, 200, 50)$ €

$g = (20, 5, 30, 10, 10, 10, 25)$ kg

$cu = (5, 10, 5, 10, 2, 20, 2)$ €/kg

Pas 1: $O_6, O_2, O_4, O_1, O_3, O_5, O_7$

Obiect	Spațiu liber	Câștig total
–	65	0
O_6	55	200
O_2	50	250
O_4	40	350
O_1	20	450
O_3	0	550

METODA DIVIDE ET IMPERA

C_1 (Divide): Pb. se poate împărți în mai multe sbpb. de ac. tip (tb. să aibă datele de intrare aprox. egale) și care au datele de intrare cu dimensiuni aprox. egale.

C_2 (Impera): Soluția unei pb. se poate obține combinând soluțiile subpb. sale

Algoritm general D.I.:

st dr
L = [5, 7, -10, 3, 4, -9, 1]

```
def divimp (L, st, dr):
    if dr-st <= 1:
        return sol.pb. direct rez.
    (else :)
    mij = (st+dr)/2
    solst = divimp (L, st, mij)
    soldr = divimp (L, mij+1, dr)
    return solutie (solst, soldr)
              ↳ funcție / expresie
```

Ex: Suma elem. dintr-o listă

```
def suma(L, st, dr):
    if st == dr:
        return L[st]  ①

  ② mij = (st+dr)//2
  ③ solst = suma(L, st, mij)
  ④ soldr = suma(L, mij+1, dr)
  ⑤ return solst + soldr
```

Apel: suma(L, 0, len(L)-1)

Complexitate

$T(n)$ = complexitatea rezolvării unei pb. având dimensiunea datelor de intrare $= n$

$n = dr - st + 1$ / apel suma $(L, 0, \underbrace{len(L)-1}_{n})$

$$T(n) = \begin{cases} 1, & n = 1 \\ 1 + T(\frac{n}{2}) + T(\frac{n}{2}) + 1, & n \geq 2 \end{cases} = \begin{cases} 1, & n = 1 \\ 2T(\frac{n}{2}) + 2, & n \geq 2 \end{cases}$$

Presupunem că $n = 2^k$

$$T(n) = T(2^k) = 2T(2^{k-1}) + 2 = 2[2T(2^{k-2}) + 2] + 2$$
$$= 2^2 \cdot T(2^{k-2}) + 2^2 + 2$$
$$\dots$$
$$= 2^k + 2^k + 2^2 + 2 = 2^k + 2 \cdot (2^k - 1) = 3 \cdot 2^k - 2$$
$$3 \cdot n - 2 = O(3n - 2) = O(n)$$