Sem10-B3
Thursday, 15 April 2021 22:18

Problema rucsacului

input: n ∈ ℕ, n ≥ 1
v[0...n-1] - natural, v[i] ∈ ℕ, i ∈ [0,n-1]
g[0...n-1] - natural, g[i] ∈ ℕ, i ∈ [0,n-1]
K - natural, K ∈ ℕ

output: f[0...n-1] (sau doar ultima valoare) — cerința 1: f[i] ∈ {0,1}, i ∈ [0,n-1]
cerința 2: f[i] ∈ {0,1}
a.î. $\sum_{i=0}^{n-1} f[i]\cdot g[i] \le K$ (validă)
$\sum_{i=0}^{n-1} f[i]\cdot v[i] \to \max$ (optim)

a) continuă

x → decizia i cel mai curând (v[i]/g[i] - max) din cele rămase
f[i] → 0 sau 1 a.î. f[i]·g[i] ≤ spațiul rămas

Subproblema: (S, K) — S: decizile rămase; K: spațiul rămas
S ⊆ {0,...,n-1}, K ∈ {0...C}

Tema de acasă alg.

(S, K) —x←i, f[i]→ (S', K') — subproblema după de x
S' = S \ {i}
K' = K - g[i]·f[i]

b) discretă

	0	1
v	25	40
g	5	10
v/g	5	4

C = 10

Programare dinamică

S, K —x←i, f[i]=1→ v[i] + [S' = S \ {i}, K' = K - g[i]] — lema de substructură optimă; max_{i ∈ S} (v[i] + castig max pt (S', K'))
S, K —f[i]=0→ castig max pt (S, K)
max(15, 15, 12) = 15
S, K —f[i]=1→ 12

f = [_ _ 1/i1 _ 1/i2]
S

f = [_ _ _ _ 0/i-1]

S = {0, ..., i-1}
K
i

f[i-1]=1 → DA: v[i-1] + [S' = {0...i-2}, K' = K - g[i-1], i' = i-2]
f[i-1]=0 → NU: [S' = {0...i-2}, K' = K, i' = i-1]

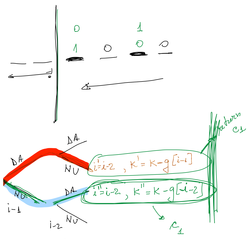
pt decizii i-1 intru doar în sum în varianta i ∈ {n, n-1, ..., 0}

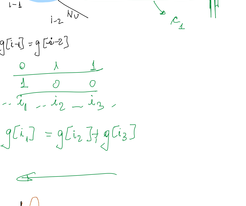
castig(K, i) = 0, dacă K = 0 sau i = 0
max(v[i-1] + castig(K - g[i-1], i-1), castig(K, i-1)), dacă g[i-1] ∈ K (K ≠ 0 și i ≠ 0)
castig(K, i-1), altfel

m[K][i] = rezultatul pt castig(K, i)
└ inițial -1

castig(K, i) — modifică g[], v[], n → g, n, v globale

```
}
if(m[K][i] != -1) return m[K][i];
if(K == 0 || i == 0){m[K][i]=0; return 0;}
max = castig(K, i-1);
if(g[i-1] <= K)
{ a = v[i-1] + castig(K-g[i-1], i-1);
  if(a > max) max = a;
}
m[K][i] = max;
return max;
}
```



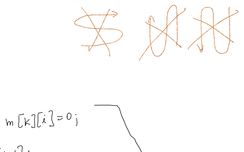


g[i-1] = g[i-2]

0	1	1
1	0	0

... i_1 ... i_2 ... i_3 ...

g[i_1] = g[i_2] ≠ g[i_3]



```
for(K = 0; K <= C; K++)
  for(i = 0; i <= n; i++)
  { if(i == 0 || K == 0) m[K][i] = 0;
    else{ max = m[K][i-1];
          if(g[i-1] <= K)
          { a = v[i-1] + m[K-g[i-1]][i-1];
            if(a > max) max = a;
          }
          m[K][i] = max;
    }
  }
return m[C][n];
```

Tema: Ex 1b

INPUT: A ⊆ ℤ

OUTPUT: DA dacă ∃ B ⊆ A, C ⊆ A, C ∩ B = ∅, C ∪ B = A
și $\sum_{b \in B} b = \sum_{c \in C} c$

NU altfel

∃ B ⊆ A a.î. (C = A \ B)

$\sum_{b \in B} b = \sum_{a \in A} a - \sum_{b \in B} b$ ⟺

$\sum_{b \in B} b = \frac{\sum_{a \in A} a}{2}$ — Asum

A → v[0, ..., A.size() - 1]
n = A.size();
→ i de la n ... 0
→ pt elementul i-1 — DA: (i-1, K - v[i-1]) — true; NU: (i-1, K) — false
{0, ..., i-2}, K' = K - v[i-1]
{0, ..., i-2}, K
(i, K) — K: suma rămasă de adunat — true

Tema:
- relația de recurență
- alg recursiv + memorizare
- alg iterativ