

**Úvod do dynamického programovania
(cvičenie)**

Broňa Brejová

29.9.2022

Problém platenia minimálnym počtom mincí

Vstup: hodnoty k mincí m_1, m_2, \dots, m_k a cieľová suma X (všetko kladné celé čísla)

Výstup: najmenší počet mincí, ktoré potrebujeme na zaplatenie X

Príklad: $k = 3, m_1 = 1, m_2 = 2, m_3 = 5, X = 13$

Odbočka: ešte matematickejšia formulácia bez slov minca, suma,...

Vstup: kladné celé čísla m_1, m_2, \dots, m_k a X

Výstup: celé číslo n a n čísel x_1, \dots, x_n , pre ktoré platia nasledujúce podmienky:

- $x_i \in \{m_1, m_2, \dots, m_k\}$ pre každé $i = 1, 2, \dots, n$
- $\sum_{i=1}^n x_i = X$
- n je najmenšie možné.

Problém platenia minimálnym počtom mincí

Vstup: hodnoty k mincí m_1, m_2, \dots, m_k a cieľová suma X (všetko kladné celé čísla)

Výstup: najmenší počet mincí, ktoré potrebujeme na zaplatenie X

Príklad: $k = 3, m_1 = 1, m_2 = 2, m_3 = 5, X = 13$

Príklad: $k = 3, m_1 = 1, m_2 = 3, m_3 = 4, X = 6$

Algoritmus pre všeobecnú sústavu k mincí m_1, m_2, \dots, m_k

$$A[i] = 1 + \min\{A[i - m_1], A[i - m_2], \dots, A[i - m_k]\}$$

```
m = [1, 3, 4]
X = 11
k = len(m)
nekonecno = math.inf
A = [0]
for i in range(1, X + 1):
    min = nekonecno
    for j in range(k):
        if i >= m[j] and A[i - m[j]] < min:
            min = A[i - m[j]]
    A.append(1 + min)
print(A)
```

Program aj s výpisom mincí

```
m = [1, 3, 4]
X = 11
k = len(m)
nekonecno = 1000000
A = [0]
B = [-1]
for i in range(1, X + 1):
    min = nekonecno
    min_minca = -1
    for j in range(k):
        if i >= m[j] and A[i - m[j]] < min:
            min = A[i - m[j]]
            min_minca = m[j]
    A.append(1 + min)
    B.append(min_minca)

while X > 0:
    print(B[X])
    X = X - B[X]
```

Dynamické programovanie vo všeobecnosti

- Okrem riešenia celého problému riešime aj menšie problémy (nazývame ich podproblémy)
- Riešenia podproblémov ukladáme do tabuľky a používame pri riešení väčších podproblémov
- Technika dynamického programovania sa používa na viacero problémov v bioinformatike