Cvičenie 08: Dynamické programovanie

Efektívne Algoritmy a Dátové Štruktúry
FMFI UK

Rekurzívne rozmieňanie mincí

```
Change(N):
  best = infinity
  for c in coins:
    best = min(best, Change(N - c))
  return best
```

Rekurzívne rozmieňanie mincí

```
Change(N):

  if N in mem:
    return mem[N]

  best = infinity
  for c in coins:
    best = min(best, Change(N - c))

  mem[N] = best

  return best
```

Rekurzívny predaj vína

Máme n fliaš vína s hodnotou $c_1 \ldots c_n$.

Ak fľašu s hodnotou c_i predáme v roku r, dostaneme za ňu $r \cdot c_i$ peňazí. Každý rok musíme predať jednu fľašu vína. Fľaše vieme predávať iba tie (dve), ktoré sú na kraji.

Koľko najviac peňazí vieme zarobiť?

Pripomeňme si riešenie:

$$D[i,j] = \max \begin{cases} D[i+1,j] + c[i] \cdot y \\ D[i,j-1] + c[j] \cdot y \end{cases}$$

Rekurzívny predaj vína

```
Wine(L, R):
  return max(
    Wine(L+1, R) + cost[L] * year,
    Wine(L, R-1) + cost[R] * year,
  )
```

↓

```
Wine(L, R):
  if (L,R) not in mem:
    mem[L,R] = max(
      Wine(L+1, R) + cost[L] * year,
      Wine(L, R-1) + cost[R] * year,
    )
  return mem[L,R]
```

Rekurzívny predaj vína (Python)

```
def Wine(L, R):
  return max(
    Wine(L+1, R) + cost[L] * year,
    Wine(L, R-1) + cost[R] * year,
  )
```

↓

```
import functools

@functools.cache
def Wine(L, R):
  return max(
    Wine(L+1, R) + cost[L] * year,
    Wine(L, R-1) + cost[R] * year,
  )
```

Memoizácia

- Alternatívna forma dynamického programovania.
- Časová zložitosť $O(mS)$, výpočet podproblému trvá $\Theta(m)$, každý z S podproblémov počítame iba raz.
- Počítame iba podproblémy, ktoré potrebujeme.
- Komplikovanejšia analýza.
- Overhead rekurzie.