

# DYNAMICKÉ PROGRAMOVÁNÍ II

---

Karel Horák, Petr Ryšavý

4. května 2016

Katedra počítačů, FEL, ČVUT

Najděte nejdelší rostoucí podposloupnost dané posloupnosti. Použijte dynamické programování. 5 8 11 13 9 4 1 2 0 3 7 10 12 6

Určete, jak lze využít nebo přizpůsobit metodu hledání nejdelší rostoucí podposloupnosti pro hledání

1. nejdelší klesající podposloupnosti,
2. nejdelší neklesající podposloupnosti,
3. nejdelší konstantní podposloupnosti,
4. nejdelší alternující posloupnosti.

---

Nápověda: v alternující posloupnosti  $a_1, a_2, \dots, a_n$  platí

$$(a_k - a_{k-1}) \cdot (a_{k+1} - a_k) < 0$$

pro všechna  $k = 2, 3, \dots, n - 1$ .

Kolika různými způsoby lze uzávorkovat součin matic?<sup>1</sup>

---

<sup>1</sup>Za různé způsoby uzávorkování považujeme pouze uzávorkování, která vedou k různým pořadím výpočtu.

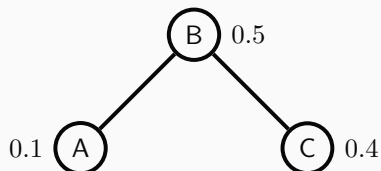
Předpokládejme, že na vypočtení součinu matic  $\mathbf{A} \times \mathbf{B}$  je zapotřebí  $r \cdot s \cdot t$  operací, kde  $\mathbf{A} \in \mathbb{R}^{r \times s}$  a  $\mathbf{B} \in \mathbb{R}^{s \times t}$ . Určete, kolik operací je třeba na výpočet součinu  $(\mathbf{A} \times \mathbf{B}) \times \mathbf{C}$  a kolik na výpočet součinu  $\mathbf{A} \times (\mathbf{B} \times \mathbf{C})$ , pokud

$$\mathbf{A} \in \mathbb{R}^{5 \times n}, \mathbf{B} \in \mathbb{R}^{n \times 4}, \mathbf{C} \in \mathbb{R}^{4 \times n}.$$

Rozměry matic **A**, **B**, **C**, **D**, **E**, jsou po řadě  
 $2 \times 5$ ,  $5 \times 3$ ,  $3 \times 6$ ,  $6 \times 2$ ,  $2 \times 4$ .

Určete metodou dynamického programování, jak uzávorkovat součin  
 $\mathbf{A} \times \mathbf{B} \times \mathbf{C} \times \mathbf{D} \times \mathbf{E}$ , aby počet operací násobení dvou čísel během  
výpočtu celého součinu byl co nejmenší. Kolik to bude operací?

Pravděpodobnost dotazu na konkrétní klíče daného BVS je uvedena u jednotlivých uzlů. Jaký je dlouhodobě průměrný počet navštívených uzlů při jednom dotazu na klíč ve stromu?

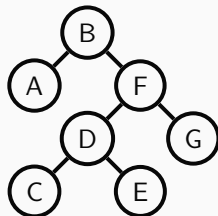
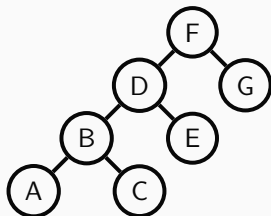


## Příklad 7

Pravděpodobnost dotazu na jednotlivé klíče v obou daných BVS je:

A : 0.10, B : 0.20, C : 0.25, D : 0.05, E : 0.10, F : 0.25, G : 0.05

Vypočtete, který strom je výhodnější pro operaci `find()`.





Určete, jak bude vypadat optimální BVS, když jej vybudujeme pro 7 klíčů s frekvencemi

E : 0.04, F : 0.05, G : 0.22, H : 0.04, I : 0.06, J : 0.05, K : 0.15.

V dané matici začneme kdekoli v prvním sloupci a dále vždy postupujeme jen ve směru SV, V, JV až kamkoli do posledního sloupce. Cena cesty je součet hodnot navštívených polí. Jaká může být nejmenší?

29	10	11	23	22	23
27	25	29	12	29	24
18	21	11	27	14	24
30	17	26	29	23	22
12	25	23	13	28	16
20	24	10	14	30	15

Najděte nejdelší společnou podposloupnost dvojice řetězců

A : 11101001000

B : 00010010111

Řešte 0/1 variantu úlohy batohu pro kapacitu 130 a pět předmětů, jejichž váhy jsou postupně 20, 30, 40, 50, 60 a ceny jsou ve stejném pořadí předmětů 11, 33, 45, 58, 65.

Jaká bude maximální cena předmětů v batohu?

Uvažte, jak se vyhnout tabulce se cca 130 sloupci a upravte postup řešení tak, aby stačilo řádově méně sloupců.

Každá zásilka ve skladu má určenou váhu a je nutno všechny zásilky naložit na dvě přistavené dodávky tak, aby se váhy nákladů na obou dodávkách lišily co nejméně.

Zdůvodněte, zda lze či nelze metodu DP pro řešení úlohy batohu využít i v této situaci.