Editační vzdálenost

$L(\alpha,\beta)$

kaule < kdaule, kule, baule

} editační operace: změna, přidání, smazání

↳ Edit. vzdálenost řetězců $\alpha, \beta :=$ min. délka posloupnosti edit. operací, které udělají z α β.

↳ Je to metrika

Operace jsou BÚNO prováděny zleva doprava.

$\alpha = a_1 \dots a_n \;, \beta = b_1 \dots b_n$

$L(\alpha,\beta)$

① Smaže a_1 $\quad 1+ L(a_2 \dots a_n, \beta)$

② změní a_1 na b_1 $\quad 1+ L(a_2 \dots a_n, b_2 \dots b_n)$

③ před a_1 vloží b_1 $\quad 1+ L(\alpha, b_2 \dots b_n)$

④ ponecháme a_1 a b_1 $(a_1 = b_1)$ $\quad 0+ L(a_2 \dots a_n, b_2 \dots b_n)$

Zkusíme 1…4, vybereme min.

↳ Takle se dá rekurzivně vyřešit a vrátit nejkratší editační vzdálenost

$L(\alpha_i \dots \alpha_n, \beta_j \dots \beta_n)$

Edit(i,j):

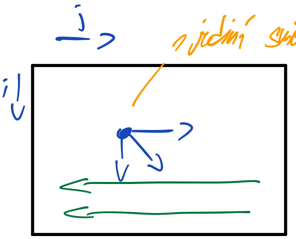
```
① Pokud $i > n$: vrátíme $m-j+1$
   Pokud $j > m$: vrátíme $n-i+1$
② $l_{vloží} \leftarrow 1+ edit(i+1, j)$
③ $l_{smaže} \leftarrow 1+ edit(i, j+1)$
④ $l_{změn} \leftarrow edit(i+1, j+1)$
⑤ Pokud $\alpha_i \neq \beta_j$ : $l_z \leftarrow l_z + 1$
⑥ Vrátíme min. $(l_v, l_s, l_z)$
```

Pomalé pro např.: stejné posloupnosti α i β

$i \in \{1 \dots n+1\}, j \in \{1 \dots m+1\}$

Celkem existuje jen $\Theta(n \cdot m)$ různých variant parametrů volání, přestože jich je zavoláno celkem víc

Použiju hashovací tabulku a budu si brát hodnoty



↗ jediné směry volání

hash. tabulka

↗ Pokud bych vyplňoval zprava odspodu, zajistím si odpovědi

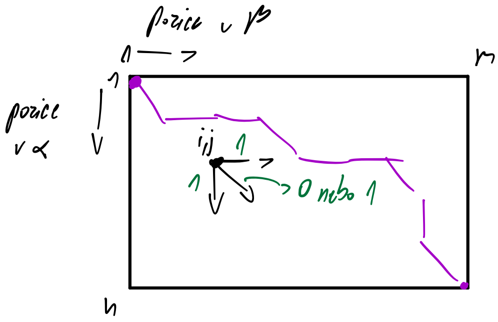
To je návod na nerekurzivní alg.

```
① Pro $j = 1 \dots m+1$:
      $T[n+1, j] \leftarrow m-j+1$
   Pro $i = 1 \dots n$:
      $T[i, m+1] \leftarrow n-i+1$
② Pro $i = n \dots 1$:
③    Pro $j = m \dots 1$:
④       $\delta = \begin{cases} 0 & \text{pokud } \alpha_i = \beta_j \\ 1 & \text{else} \end{cases}$
⑤       $T[i,j] = \min(1+T[i+1,j], 1+T[i,j+1], \delta + T[i+1,j+1])$
```

Čas. složitost $= \Theta(n \cdot m)$

Prostor. složitost $= \Theta(n \cdot m)$

Grafový pohled:



nejkratší
Cesta z (1,1) do (n,m)
nejkratší
posl. editací, udělá α z β.

Vytvoříme graf → nejkratší cesta → editační vzdálenost

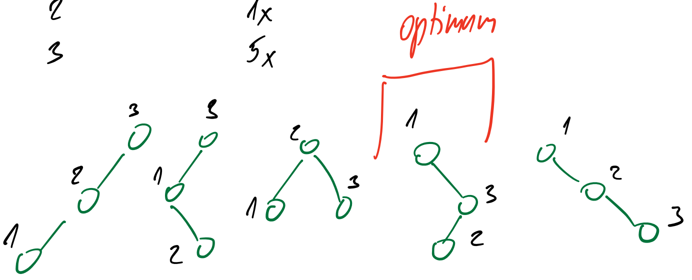
$\Theta(n \cdot m)$ v DAGu $\Theta(n \cdot m)$

} $\Theta(n \cdot m)$

Optimální BST

↗ zohlednění četnosti dotazů

Na 1 se ptám 10x
2 1x
3 5x



Cena: 37 28 31 23 27

↗ vyvážený není nejlepší!

Problém:
Dány klíče $x_1 - x_n$
a váhy $w_1 - w_n \in \mathbb{N}$

Pro BST T na $x_1 - x_n$: hloubky $h_1 - h_n$

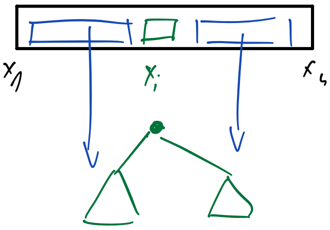
$h_i := \#$ vrcholů na cestě kořen → x_i

cena

$C(T) = \sum_i x_i \cdot w_i$

Cíl: najít min. $(C(T))$ pro nějaký BST T na $x_1 - x_n$.

Co kdyby v kořeni opt. stromu bylo x_i



Jestli kořen optimální, pak podstromy taky optimální

$OPT(x_1 - x_n) =$
$OPT(x_1 - x_{i-1})$
$+ OPT(x_{i+1} - x_n)$
$+ \sum_{j=1}^{n} w_j$

→ Kořen ale neznáme, tak dosadíme každý vrchol postupně za kořen

opt. cena BST pro $x_l - x_p$

```
OptBST(l,p):
    1. Pokud $l > p$: return 0.
    2. $\min(C_l - C_p) + \sum_{i=l}^{p} w_i$
       kde $C_i := OptBST(l,i-1) + OptBST(i+1,p)$
```

Je to pomalé

$l, p \in \{1 - n+1\}$ → jen $O(n^2)$ podproblémů
└ zavedeme hashování
↘ v čase $O(n)$ } celkem $O(n^3)$

Nahradíme rekurzi cyklem … od nejkratších intervalů k nejdelším

Bez rekurze:

```
Pro $\ell = 1 \ldots n$ : $T[\ell, \ell-1] \leftarrow 0$
Pro $d = 1 \ldots n$ :            d = délka intervalu
    Pro $\ell = 1 \ldots n-d+1$ :    $\ell$ := levý okraj
        $p \leftarrow \ell + d - 1$        p := pravý okraj
        $T[\ell, p] \leftarrow \min(C_\ell \ldots C_p) + \sum_{i=\ell}^{p} w_i$
            ↳ $C_i := T[\ell, i-1] + T[i+1, p]$
Vrátíme $T[1, n]$
```

čas: $O(n^3)$

prostor: $O(n^2)$

Zapamatujeme si, kde se došlo minimum, to je kořen pro daný interval

Cenu máme, ale jak vypadá strom?

→ Pak umím vytvořit triviálně strom

Dynamické programování – obecně

Systém podproblémů – stavy
a závislostí mezi nimi → tvoří DAG

Procházíme stavy v topologickém pořadí.

Mějme orientovaný graf s vrcholy $\{1 \ldots n\}$ a matici délek hran $L \in \mathbb{R}^{n \times n}$: $L_{ij} = \begin{cases} \text{délka hrany } (i,j) \\ +\infty \text{ pokud hrana neexistuje} \end{cases}$

Chceme matici vzdáleností $D \in \mathbb{R}^{n \times n}$:

$$D_{ij} = \text{délka nejkratší cesty z } i \text{ do } j.$$

Umíme:

$n \times$ Dijkstra : $\Theta(n^3)$

$n \times$ Bellman Ford : $\Theta(n^4)$

Ukážeme:

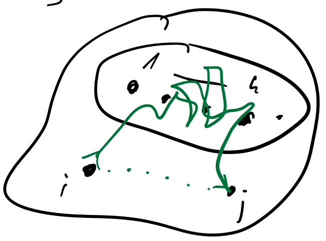
Floydův-Warshallův alg. taky $O(n^3)$, ale triviální

Df: D_{ij}^{k} := délka nejkratšího sledu z i do j, jehož vnitřní vrcholy leží v $\{1 \ldots k\}$

↳ matice $D^0, D^1, \ldots D^n$

matice L ↗ ? (D^0)

$= D$ (D^n)

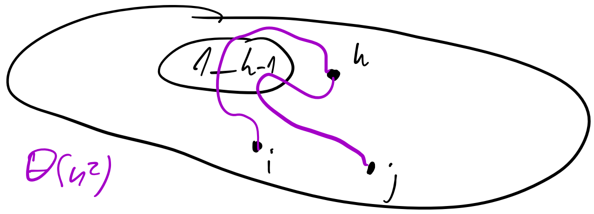


Výpočet D^k z D^{k-1}

$$D^k_{ij} := \min\left(D^{k-1}_{ij},\; D^{k-1}_{ik} + D^{k-1}_{kj}\right)$$

k nepoužito — k použito (Bino) 1x

$\Theta(n^2)$



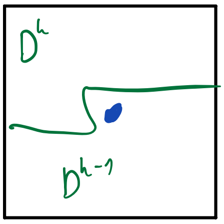
n kroků: $D^0 \to D^1 \to D^2 \to \dots \to D^n$ čas $\Theta(n^3)$

Nevýhoda:
paměť $\Theta(n^3)$

Stačí si pamatovat D^k a D^{k-1} $\Rightarrow \Theta(n^2)$

Nebo můžu přepisovat matici na místě

$D^{k-1}_{ik} = D^k_{ik}$
$D^{k-1}_{kj} = D^k_{kj}$
přepis neškodí



Celkem:

```
Pro k = 1 — n:
    Pro i = 1 — n:
        Pro j = 1 — n:
            $D_{ij} = \min(D_{ij}, D_{ik} + D_{kj})$
```

?
Očividně $O(n^3)$