

## Cvičení 9 – Trie, intervalové stromy

**Úloha 1** Sestrojte (složenou) datovou strukturu, která v zadaném čase zvládne následující operace:

- $\text{INIT}()$  inicializuje datovou strukturu v  $\mathcal{O}(1)$ ,
- $\text{INSERT}(X)$  vloží prvek  $X$  pokud v datové struktuře ještě není v  $\mathcal{O}(\log n)$ ,
- $\text{DELETE}(X)$  smaže  $X$  pokud je ve struktuře v  $\mathcal{O}(\log n)$ ,
- $\text{DELETEINPLACE}(I)$  smaže  $I$ -tý přidaný prvek v  $\mathcal{O}(\log n)$ ,
- $\text{GETPLACE}(X)$  vrátí číslo  $I$ , kolikátý byl vložený prvek  $X$  v  $\mathcal{O}(\log n)$ .

### $(a, b)$ -stromy

**Úloha 2** (*Volba parametrů*) Určete, jak závisí složitost operací s  $(a, b)$ -stromy na parametrech  $a$  a  $b$ . Z toho odvoďte, že se nikdy nevyplatí volit  $b$  výrazně větší než  $2a$ .

**Úloha 3** (*Jednosměrné operace*) Navrhněte úpravu operací  $\text{INSERT}$  a  $\text{DELETE}$  do  $(a, b)$ -stromů, aby operace procházely stromem pouze od kořene dolů a nebylo nutné jít zpět. Předpokládejte  $b \geq 2a$ .

### Trie

**Úloha 4** (*Třídění pomocí trie*) Vymyslete, jak pomocí písmenkového stromu setřídít posloupnost řetězců v čase lineárním vzhledem k součtu jejich délek. Porovnejte s algoritmem přihrádkového třídění. (Uvažujte konstantně velkou abecedu.)

**Úloha 5** (*Rýmy*) Navrhněte datovou strukturu pro básníky, která si bude pamatovat slovník a bude umět hledat rýmy. Tedy pro libovolné zadané slovo najde jiné slovo ve slovníku, které má se zadaným co nejdelší společný suffix (koncovku).



Jakub Komárek

[komarek+ads1@iuuk.mff.cuni.cz](mailto:komarek+ads1@iuuk.mff.cuni.cz)

<https://jakoma02.cz/teaching/ls24/ads1/>

**Úloha 6** (*Přesmyčky*) Jak reprezentovat slovník (množinu slov), abyste uměli rychle vyhledávat všechny přesmyčky zadaného slova?

### Intervalové stromy

**Úloha 7** (*Druhý nejmenší*) Naučte intervalový strom zjistit druhý nejmenší prvek v zadaném intervalu.

**Úloha 8** (*Nejbližší větší napravo*) Naučte intervalový strom zjistit nejbližší prvek, který leží napravo od zadaného listu a obsahuje větší hodnotu. Jinými slovy: na vstupu dostaneme index  $i$  v posloupnosti a chceme najít nejmenší  $j > i$  takové, že  $x_j > x_i$ .

**Úloha 9** (*Intervalový update*) Naučte (součtový) intervalový strom operaci  $\text{UPDATE}(i, j, \delta)$ , která k hodnotám  $x_i, x_{i+1}, \dots, x_j$  reprezentované posloupnosti přičte  $\delta$ . Jelikož těchto hodnot může být až  $\Omega(n)$  a my chceme logaritmickou časovou složitost, stačí jen upravit strom tak, aby i po provedení této operace vracel správné výsledky pro dotazy na součet intervalu.

**Úloha 10** (*Jeřáb*) Jeřáb se skládá z  $n$  ramen spojených klouby. Pro jednoduchost si ho představíme jako lomenou čáru v rovině. První úsečka je fixní, každá další je připojena kloubem na svou předchůdkyni. Koncový bod poslední úsečky hraje roli háku. Navrhněte datovou strukturu, která si bude pamatovat stav jeřábu a bude nabízet operace „otoč  $i$ -tým kloubem o úhel  $\alpha$ “ a „zjisti aktuální pozici háku“.



Jakub Komárek

[komarek+ads1@iuuk.mff.cuni.cz](mailto:komarek+ads1@iuuk.mff.cuni.cz)

<https://jakoma02.cz/teaching/l24/ads1/>