

1 Trie:

1.1 Třídění pomocí trie

Vymyslete, jak pomocí písmenkového stromu setřídit posloupnost řetězců v čase lineárním vzhledem k součtu jejich délek. Porovnejte s algoritmem přihrádkového třídění.

1.2 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).

1.3 Přesmyčky.

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

2 LLRB:

2.1 hloubka LLRB.

Spočítejte přesně, jaká může být minimální a maximální hloubka LLRB stromu s n klíči (přesně znamená určit i multiplikativní konstanty; aditivní konstanty určit přesně nemusíte). Porovnejte maximální hloubku s AVL stromem.

3 Intervalové stromy:

3.1 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 $x_j > x_i$.

3.2 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 α “ a „zjistí aktuální pozici háku“.

Bonusové úlohy:

Lexikograficky nejmenší rým.

Upravte datovou strukturu na hledání nejlepších rýmů, aby v případě, že nejlepších rýmů je více, vypsal lexikograficky nejmenší z nich.