

Trie: Slovníkový/písmenkový strom.

Terminální znaky: Pokud chceme mít v trii všechny konce slov jasně vyznačené, můžeme to udělat tak, že na konec každého slova přidáme nový znak \$, který se jinde v abecedě nevyskytuje. Když pak nad těmito slovy postavíme trii, bude nám jasné, které vrcholy označují konce slov (ty, které mají potomka \$).

Komprimovaná trie: zakomprimujeme všechny vrcholy z trie, které mají stupeň dva. Hrany mají nyní popisky řetězce, ne jen znaky.

Suffixový strom pro slovo σ : komprimovaná trie nad slovníkem obsahujícím všechny suffixy σ .

PŘÍKLAD PRVNÍ Pro slovo σ délky n nejdřív zkonstruujeme obyčejnou trii obsahující všechny suffixy slova σ . Kolik paměti taková trie v nejhorším případě zabere?

Pak trii zkomprimujeme. Jaká je nejhorší paměťová složitost této zkomprimované trie? (Velikost může záviset na tom, jak budete mít uloženy popisky na hranách; vyberte nejlepší reprezentaci.)

Věta: Existuje algoritmus, který pro zadané slovo σ délky n zkonstruuje suffixový strom v čase $O(n)$.

PŘÍKLAD DRUHÝ První krok se suffixovým stromem: jak najít lexikograficky nejmenší rotaci jednoho slova v lineárním čase?

PŘÍKLAD TŘETÍ Na přednášce jste strávili nějaký ten čas návrhem a analýzou algoritmu, který v čase $O(m+n)$ najde v seně délky m všechny výskyty jehly délky n .

Zkuste použít suffixový strom k tomu, abyste zvládli to samé. Stačí vypsát počet všech výskytů.

PŘÍKLAD ČTVRTÝ Ukažme si sílu suffixových stromů na příkladu, který už známe: na problému NEJDELŠÍ SPOLEČNÝ PODŘETĚZEC. Pro připomenutí:

Na vstupu máme řetězec α délky a a β délky b a chceme najít co nejdelší řetězec, který je nejdelším podřetězcem jak α , tak β . (Nejdelších podřetězců může být víc, stačí nám jeden libovolný.)

My jsme doposud viděli algoritmus, který pro řetězec délky a a řetězec délky b našel nejdelší společný podřetězec v čase $O(a \cdot b)$. Použijte suffixové stromy a vyřešte stejnou úlohu v čase $O(a+b)$.

Co ještě zvládne suffixový strom vyřešit v lineárním čase?

- Nejdelší podslovo, co je palindrom,
- nejdelší opakující se podslovo,
- Burrows-Wheelerovu transformaci,
- a skoro všechno ostatní.

„Největší nevýhoda“ suffixových stromů: zkoušející/cvičící někdy chtějí slyšet konstrukci, pokud je chcete používat na zkoušce/písemce.

PŘÍKLAD PÁTÝ Jedna z konstrukcí suffixových stromů používá ekvivalentní datovou strukturu, přesněji ekvivalentní dvojici polí: *suffixového pole* a *pole společných prefixů*.

D(Suffixové pole SA): Lexikograficky seřazené pole všech suffixů slova σ . Aby se ušetřilo na paměti, tak jsou v poli uloženy jen čísla od 0 do n : začátky suffixů.

D(Pole společných prefixů LCP): Pole je postaveno jako doplněk suffixového pole. V $LCP[i]$ budeme mít uloženou délku nejdelšího společného prefixu $SA[i]$ a $SA[i+1]$.

1. Ukažte, že se dá ze suffixového stromu v lineárním čase postavit suffixové pole i pole LCP.
2. Zkuste vymyslet i rekonstrukci suffixového stromu ze suffixového pole a pole společných prefixů.