2. CVIČENÍ Z ADS2

Vyhledávání jehel v seně

Algoritmus **Knuth-Morris-Pratt** v deseti slovech:

„Hledej po znacích, vracej se zpět chytřeji: zpětné hrany! Bootstrapping.“

Algoritmus **Aho-Corasicková** v deseti slovech:

„Postav trii, zpětné hrany jako KMP. Zkratky do nejdelších suffixů.“

Opakování óčkové notace:

$f \in O(g)$	f je asymptoticky stejná nebo **menší** než g
$f \in \Omega(g)$	f je asymptoticky stejná nebo **větší** než g
$f \in \Theta(g)$	f je asymptoticky stejná, jako g
$f \in \omega(g)$	f je asymptoticky ostře větší, než g (příklad: $n \log n \in \omega(n)$)
$f \in o(g)$	f je asymptoticky ostře menší, než g (příklad: $\sqrt{n} \in o(n)$)

Příklad první Ještě jednou k rotacím: Nalezněte algoritmus, který pro slova γ, δ na vstupu rozhodne, je-li γ rotací δ nebo nikoli.

Příklad druhý Na vstupu máme jen jedno slovo σ, a hledáme jeho nejdelší prefix, který je zároveň i jeho suffixem. Jak na to?

Příklad třetí Vyřešme teď jeden klasický řetězcový problém, který nesouvisí s hledáním jehel v seně: takzvaný Nejdelší společný podřetězec. 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.)

1. Ukažte (jakkoli hloupě chcete), že problém je polynomiálně řešitelný.
2. Potom zkusíme najít řešení v $O(a \cdot b)$. Trénink: pokud vám prozradím tuto složitost, jaké typy algoritmů byste zkusili?

Zajímavý fakt na konec: problém jde řešit i v $O(a + b)$. Takové řešení ale je nad rámec ADS2.

Hledání více jehel v jednom seně: Na vstupu máme seno σ a několik jehel $\iota_1, \iota_2, \ldots, \iota_k$. Celková délka sena je s, celková délka všech jehel $j_1 + j_2 + \ldots + j_k = J$. Našim úkolem je vypsat dvojici (c, i), kdykoli na pozici c v seně končí jehla ι_i. Inspirováni KMP bychom chtěli složitost $O(s + J)$.

Příklad čtvrtý Ukažte, že takhle snadno to nepůjde:

Nalezněte příklad jehel a sena, v němž je asymptoticky více než lineární počet výskytů. Přesněji řečeno ukažte, že pro každé n existuje vstup, v němž je součet délek jehel a sena $\Theta(n)$ a počet výskytů je $\omega(n)$ (čili je jich asymptoticky více).

Dobrá, upravujeme svůj požadavek: chtěli bychom složitost $O(s + J + V)$, kde V je počet všech výskytů.

Příklad pátý Uvažujme naivní algoritmus verze 1, který pracuje s trií, ale nepoužívá zkratkové hrany a vždy projde po zpětných hranách až do kořene. Ukažte, že tento algoritmus je asymptoticky pomalejší než náš požadavek.

Příklad šestý Druhý snadný způsob, jak si poradit s hlášením výskytů, je předpočítat si pro každý vrchol trie v množinu $M(v)$ indexů slov k ohlášení. Dokažte, že pro slovník velikosti J může být součet $M(v)$ v celé trii dokonce $\omega(J)$ – takže předpočítat je v čase $O(J)$ nestihneme.

(Pozor na to, že nám jde o součet velikostí množin, ne součet všech délek slov v těchto množinách. Když má $M(v)$ tři slova délky sto, tak $|M(v)|$ je stále 3.)