Pokročilé vyhledávání v textu

Pojmy: trie, Aho-Corasicková

Příklad 1: Ukažte, že v automatu A.-K. není možné si pro stavy pamatovat všechny jehly k vypsání, a dosáhnout lineárního času.

Příklad 2: Navrhněte úpravu algoritmu, která pouze počítá počet výskytů. Chceme časovou složitost nezávisející na počtu výskytů.

Příklad 3: Jak upravit automat, pokud povolíme neomezeně velké abecedy? (velikost abecedy je závislá na vstupu)

Příklad 4: Navrhněte algoritmus, který v matici $n \times n$ hledá vzory $k \times k$ v lineární čase. Upravte pro vyhledávání více vzorů.

Pojmy: Rabin-Karp

Příklad 5: Opakování časové analýzy Rabin-Karpova algoritmu.

Příklad 6: Najděte v textu nějaký podřetězec délky k, který se vyskytuje alespoň dvakrát.

Pokročilé vyhledávání v textu

Pojmy: trie, Aho-Corasicková

Příklad 1: Ukažte, že v automatu A.-K. není možné si pro stavy pamatovat všechny jehly k vypsání, a dosáhnout lineárního času.

Příklad 2: Navrhněte úpravu algoritmu, která pouze počítá počet výskytů. Chceme časovou složitost nezávisející na počtu výskytů.

Příklad 3: Jak upravit automat, pokud povolíme neomezeně velké abecedy? (velikost abecedy je závislá na vstupu)

Příklad 4: Navrhněte algoritmus, který v matici $n \times n$ hledá vzory $k \times k$ v lineární čase. Upravte pro vyhledávání více vzorů.

Pojmy: Rabin-Karp

Příklad 5: Opakování časové analýzy Rabin-Karpova algoritmu.

Příklad 6: Najděte v textu nějaký podřetězec délky k, který se vyskytuje alespoň dvakrát.

Úkol 1 - maticový Rabin-Karp

Na vstupu máme matici A (seno) velikosti $n \times n$ a vzorovou matici B (jehlu) velikosti $k \times k$. Navrhněte algoritmus založený na algoritmu Rabin-Karp, který rozhodne zda A obsahuje B jako (souvislou) podmatici. Analyzujte časovou a prostorovou složitost.

Obdobně jako u vyhledávání v textu budeme chtít algoritmus, jehož časová složitost bude analogicky lepší než složitost naivního algoritmu. A stejně jako u vyhledávání v textu předpokládáme, že prvky matice jsou z nějaké konečné abecedy fixní velikosti.

Pokud zabloudíte při tvorbě návrhu či při analýze vašich algoritmů, na stránce `http://pruvodce.ucw.cz/` můžete najít *Průvodce labyrintem algoritmů*, kde je vše kolem vyhledávání v textu i hashování podrobně popsáno.

Úkol 1 - maticový Rabin-Karp

Na vstupu máme matici A (seno) velikosti $n \times n$ a vzorovou matici B (jehlu) velikosti $k \times k$. Navrhněte algoritmus založený na algoritmu Rabin-Karp, který rozhodne zda A obsahuje B jako (souvislou) podmatici. Analyzujte časovou a prostorovou složitost.

Obdobně jako u vyhledávání v textu budeme chtít algoritmus, jehož časová složitost bude analogicky lepší než složitost naivního algoritmu. A stejně jako u vyhledávání v textu předpokládáme, že prvky matice jsou z nějaké konečné abecedy fixní velikosti.

Pokud zabloudíte při tvorbě návrhu či při analýze vašich algoritmů, na stránce `http://pruvodce.ucw.cz/` můžete najít *Průvodce labyrintem algoritmů*, kde je vše kolem vyhledávání v textu i hashování podrobně popsáno.