Úloha č. 1
Knihovna

Odpověz Sfinze! 10 b

Tato úloha je vyhodnocována automaticky. Je potřeba, aby výstup programu ***přesně*** *korespondoval se specifikací výstupu níže. Jak odevzdávat tento typ úloh se můžeš dočíst na webových stránkách FIKSu pod záložkou „Jak řešit FIKS“.*

Hlavním cílem úlohy knihovna bylo zjistit počet unikátních slov a počet výskytů každého slova v zadaném textu. Něco takového lze poměrně snadno implementovat pomocí datové struktury tabulka (slovník, mapa), pro kterou je typické, že prvek tabulky je tvořen dvojicí klíč a hodnota. V našem případě bude za klíč sloužit nalezené slovo, jako hodnota pak počet výskytů daného slova.

Nyní již známe vše důležité pro vytvoření algoritmu řešícího zadanou úlohu. Nejdříve si vezmeme prvních k znaků ze zadaného textu a do tabulky si poznamenáme, že se slovo v textu vyskytlo právě jednou. Nyní ze slova odstraníme první znak a na jeho konec přidáme $k+1$-ní znak ze zadaného textu. Dále se podíváme do tabulky, zda-li se v ní již slovo nenachází. Pokud ano, zvýšíme počet výskytů o jedna, v opačném případě slovo do tabulky vložíme a poznamenáme si 1 výskyt.

Předchozí algoritmus byl dostačující pro lehčí část úlohy, težší část si žádala drobné optimlizace, které sloužily k ušetření paměti. Docházelo-li totiž k používání celého slova jako klíče, znamenalo to, že například pro $k = 10^5$ a $N = 10^6$ bylo potřeba, za předpokladu, že každé slovo je jiné a 1 znak zabere právě 1 byte, neuvěřitelných $10^5 * (10^6 - 10^5 - 1) = 9 \cdot 10^{10}$ bytů, což je asi 90 GB paměti. To má, dokonce i v dnešní době, jen málokterý osobní počítač.

Bylo tedy třeba najít způsob, jak neukládat celé slovo, ale zároveň vědět o tom, že se konkrétní v textu již vyskytlo, a moci ukládat počet jeho výskytů. Řešením tohoto problému je technika zvaná —hešování—.

Dříve uvedený algoritmus pak stačí zmodifikovat jednodušše tak, že v tabulce nebudeme udržovat dvojici slovo–výskyt, ale heš–výskyt. Potom bude potřeba, v závislosti na velikosti heše, přibližně pouhé desítky MB paměti.