Algoritmy a grafy 1 (BI-AG1), Cvičení č. 8

Rozděl a panuj, rekurze

Paralelka 104, Úterý 16:15-17:45

Cvičící: Šimon Lomič
lomicsim@fit.cvut.cz

Informace: lomicsim.github.io

Fakulta informačních technologií
České vysoké učení technické v Praze
https://courses.fit.cvut.cz/BI-AG1

(Verze dokumentu: 26. 11. 2018 16:38)

8.1 Hanojské věže

```
Algo Hanoj(n disků; z A, na B, pomocí C)
(1) Pokud n = 1: přesuň disk 1 z A na B
(2) Jinak:
(3)     Hanoj(n − 1; A, C, B) //horních n − 1 na C
(4)     Přesuň disk n z A na B
(5)     Hanoj(n − 1; C, B, A) //horních n − 1 na B
```

- Algoritmus Hanoj pro n disků provede $2^n - 1$ přesunů. (Důkaz v přednášce)

Cvičení:

(a) Dokažte, že algoritmus Hanoj je optimální (neexistuje algoritmus, který by použil méně přesunů).

(b) Modifikujte algoritmus tak, aby nikdy nepřesouval disk z tyče A na tyč B ani zpět, určete počet přesunů. (**1/2 b**)

8.2 Mergesort

Algoritmus MergeSort$(a_1, \ldots, a_n)$
(1) Pokud $n = 1$: vrať jako výsledek $b_1 = a_1$ a skonči
(2) $x_1, \ldots, x_{\lfloor n/2 \rfloor} :=$ MergeSort$(a_1, \ldots, a_{\lfloor n/2 \rfloor})$
(3) $y_1, \ldots, y_{\lceil n/2 \rceil} :=$ MergeSort$(a_{\lfloor n/2 \rfloor + 1}, \ldots, a_n)$
(4) Vrať $b_1, \ldots, b_n :=$ Merge$(x_1, \ldots, x_{\lfloor n/2 \rfloor}; y_1, \ldots, y_{\lceil n/2 \rceil})$

Cvičení:

(a) Jaké vlastnosti má Mergesort (datová citlivost, out/in-place, stabilita)?

(b) Navrhněte algoritmus, který setřídí spojový seznam s pomocnou pamětí velikosti $\mathcal{O}(1)$.

(c) Jak se změní časová složitost, pokud nebudeme posloupnost dělit na poloviny, ale na třetiny? A jak bude vypadat při dělení na k-tiny?

8.3 Karacubův algoritmus

`Karacuba`(n, n-ciferná čísla x a y)
(1) Pokud $n \le 1$: vrať xy a skonči
(2) $k = \lfloor n/2 \rfloor$
(3) $a := \lfloor x/10^k \rfloor$
(4) $b := x \bmod 10^k$
(5) $c := \lfloor y/10^k \rfloor$
(6) $d := y \bmod 10^k$
(7) $p :=$ `Karacuba`$(\lceil n/2 \rceil, a, c)$
(8) $q :=$ `Karacuba`$(\lfloor n/2 \rfloor, b, d)$
(9) $r :=$ `Karacuba`$(\lceil n/2 \rceil + 1, a + b, c + d)$
(10) Vrať $p \cdot 10^n + (r - p - q) \cdot 10^k + q$

Cvičení: Dokažte, že Karacubův algoritmus lze implementovat tak, aby používal pouze $\mathcal{O}(n)$ paměti.

8.4 QuickSelect

```
QuickSelect(x_1,...,x_n; k)
(1) Pokud n = 1: vrať x_1 a skonči
(2) p := některý z prvků x_1,...,x_n (pivot)
(3) L := prvky z x_1,...,x_n, které jsou menší než p
(4) P := prvky z x_1,...,x_n, které jsou větší než p
(5) S := prvky z x_1,...,x_n, které jsou rovny p
(6) Pokud k ≤ |L|: vrať QuickSelect(L, k)
(7) Jinak pokud k ≤ |L| + |S|: vrať p
(8) Jinak: vrať QuickSelect(P, k − |L| − |S|)
```

Cvičení:

(a) Proč navrhujeme volit za pivota prvek $x_{\lfloor N/2 \rfloor}$, a ne třeba x_1, x_N?

(b) Proč nevolíme jako pivota aritmetický průměr posloupnosti?

(c) Určete časovou složitost, pokud za pivota zvolíme vždy prvek který leží v prostředních šesti osminách seřazené posloupnosti.

8.5 Rozděl a panuj

Mějme dlouhý kabel, z jehož obou konců vystupuje po n drátech. Každý drát na levém konci je propojen s právě jedním na konci druhém a my chceme zjistit, který s kterým. K tomu můžeme používat následující operace:

(1) přivést napětí na daný drát na levém konci,

(2) odpojit napětí z daného drátu na levém konci,

(3) změřit napětí na daném drátu na pravém konci.

Navrhněte algoritmus, který pomocí těchto operací zjistí, co je s čím propojeno.

8.6 Hledání mediánu

Jsou dány dvě pole $A = (a_1, a_2, \ldots, a_n)$ a $B = (b_1, b_2, \ldots, b_m)$.
Navrhněte algoritmus, který nalezne medián sjednocení těchto polí.

8.7 Domácí úkol (0.5 b)

Inverze v posloupnosti $A = (a_1, \ldots, a_n)$ říkáme každé dvojici (i, j) takové, že $i < j$ a současně $a_i > a_j$. Vymyslete algoritmus, který spočítá, kolik daná posloupnost obsahuje inverzí.

Úkol odevzdejte na příštím cvičení (případně e-mailem).