

Cvičení 12

Úloha 1. Rozmyslete, že QUICKSORT, QUICKSELECT i LINEARSELECT jdou implementovat *in-place*, tj. přesouváním prvků v rámci pole bez použití paměti navíc.

Úloha 2. (alternativní pivoty)

Co kdybychom v algoritmu QUICKSELECT za pivot volili místo skoromediánu

- (a) „skoroskoromedián“, který leží v prostředních šesti osminách vstupu,
- (b) aritmetický průměr?

Úloha 3. (k -LINEARSELECT)

Fungovali by pro LINEARSELECT i jiné skupiny prvků než pětky? Jak to dopadne pro trojice? Jak pro sedmice?

Úloha 4. Jsou dány dvě vzestupně seřazené n -prvkové posloupnosti. Navrhněte algoritmus, který najde *medián sjednocení* obou posloupností v sublineárním čase.

Úloha 5. (medián komprimované posloupnosti)

Mějme „program na výrobu posloupnosti“ složený z n instrukcí typu „prvek x_i vypiš c_i -krát“. Navrhněte algoritmus pro nalezení mediánu takto vypsané posloupnosti. Časová složitost by měla být lineární v samotném n .

Úloha 6. Uvažme *nevyvažovaný* binární vyhledávací strom, do kterého náhodně postupně vložíme množinu $[n]$. V každém kroku náhodně zvolíme jeden z ještě nepřidaných prvků a vložíme jej do BVS.

Dokažte, že průměrná hloubka takového stromu bude $\mathcal{O}(\log n)$.

(*Jak úloha souvisí s Quicksortem?*)

Pravděpodobnostní rébusy

Úloha 7. Mějme počítač, jehož náhodným generátorem je ideální mince (každá strana padá s pravděpodobností $\frac{1}{2}$). Jak s takovou mincí generovat rovnoměrně náhodně čísla od 1 do n na co nejméně hodů (v průměru)?

Úloha 8. Uvažme, že máme minci, jejíž chování neznáme. Hlava padá s pravděpodobností p , orel s $(1-p)$, ale my p neznáme. Jak s takovou mincí simulovat ideální minci?