

Homework 1

Jakub Senko, Štefan Uherčík

13. marca 2014

PRÍKLAD 1

Nech $X = [x_1, x_2, x_3, \dots, x_n]$ je pole čísel a platí že $\forall x, y \in X : x \neq y$.
Každému $x_i \in X$ je priradené číslo w_i , pre ktoré platí:

$$\begin{aligned} w_i &> 0 \\ \sum_{i=0}^n w_i &= 1 \end{aligned} \tag{0.1}$$

Optimálny prvok postupnosti je číslo x_k pre ktoré platí:

$$\begin{aligned} \sum_{x_i < x_k} w_i &< \frac{1}{2} \\ \sum_{x_i > x_k} w_i &\leq \frac{1}{2} \end{aligned} \tag{0.2}$$

Problémom je návrh algoritmu ktorý rieši nájdenie optimálneho prvku s časovou zložitou $\Theta(n)$ a poskytnutie dokazu jeho korektnosti a zložitosti.

Navrhovane riesenie je modifikovany algoritmus *Quick Select* ktorý rieši problém nájdenia medianu v poli čísel. Tento algoritmus má obecnú zložitost $\mathcal{O}(n^2)$ pri nevhodnej voľbe pivotu, avšak pomocou procedury *Median of Medians* je možné najst dostatočne dobrý pivot na to, aby mal algoritmus vždy lineárnu zložitost. *Quick Select* je popísaný v nasledujúcom texte iba neformálne, s odkazom na relevantné zdroje s dôkazom zložitosti. Zadaná úloha je vyriešená ukázaním redukcie problému nájdenia optimálneho prvku na

problem riešený algoritmom *Quick Select + Median of Medians* [1] a dokazom, že táto procedura je vykonateľná v konštantom case. Výsledná zložitosť je teda $\mathcal{O}(n)$.

Quick Select

Vráti n -ty najmenší prvok pola, rekurzívne hľadá v časti ohraničenej indexami *left* a *right* (vrátane).

```
1: function QUICK_SELECT(list, left, right, n)
2:   pivotIndex  $\leftarrow$  RANDOM_BETWEEN(left, right)
3:   pivotIndex  $\leftarrow$  PARTITION(list, left, right, pivotIndex)
4:   if n = pivotIndex then
5:     return list[n]
6:   else if n < pivotIndex then
7:     return QUICK_SELECT(list, left, pivotIndex - 1, n)
8:   else
9:     return QUICK_SELECT(list, pivotIndex + 1, right, n)
10:  end if
11: end function
```

Quick Select beží v $\mathcal{O}(n^2)$.

Ako by bolo možné deterministicky vybrať dobrý pivot tak, aby bol výsledný algoritmus vždy lineárny.

Median Of Medians

Median of medians: http://www.youtube.com/watch?v=QAbv_4ndfo4&list=PLLH73N9cB21W1TZ6zz1dL

```
1: function MEDIAN_OF_MEDIANS(list)
2:   groups  $\leftarrow$  SPLIT_INTO_GROUPS_OF(list, 5)
3:   medians  $\leftarrow$  NEW_LIST
4:   for all group  $\in$  groups do
5:     median  $\leftarrow$  QUICK_MEDIAN(group)
6:     ADD(medians, median)
7:   end for
8:   size  $\leftarrow$  SIZE(medians)
9:   return QUICK_SELECT(medians, 0, size,  $\frac{n}{2}$ )
10: end function
```

Rozdelenie do skupín po 5 je $\mathcal{O}(n)$ ale nájdenie medianu z konštantného počtu prvkov je $\mathcal{O}(1)$. Nevieme zložitosť funkcie `quick_select`, preto skombinujeme algoritmy a spočítame algoritmu `quick_select_MoM`, ktorý vznikne nahradením

Redukcia: vyber pivot a spočítaj sumu na oboch stranách a rekurzuj do väčšej strany $[x_1, x_2, \dots, x_j, x_{j+1}, \dots, x_n]$ pokiaľ je suma na jednej strane väčšia ako $1/2$, hľadaný prvok bude na tej strane + ešte si treba pamätať súčet druhej strany

PRÍKLAD 2

0.1 EXAMPLE OF LIST (3*ITEMIZE)

- First item in a list
 - First item in a list
 - * First item in a list
 - * Second item in a list
 - Second item in a list
- Second item in a list

0.2 EXAMPLE OF LIST (ENUMERATE)

1. First item in a list
2. Second item in a list
3. Third item in a list

«««< Updated upstream

PRÍKLAD 3

PRÍKLAD 4

Tvrdenie 1: Ľubovoľná postupnosť n operácií INSERT a MIN-ALL má zložitosť $O(n)$.

Uvažujme prirodzené čísla n, k a l , pre ktoré platí $n=k+l$ (n vyjadruje počet operácií)

$$l = \begin{cases} \frac{n}{2} & \text{ak } n \text{ je párne} \\ \frac{n-1}{2} & \text{ak } n \text{ is nepárne} \end{cases}$$

$$l = \begin{cases} \frac{n}{2} & \text{ak } n \text{ je párne} \\ \frac{n-1}{2} + 1 & \text{ak } n \text{ is nepárne} \end{cases}$$

Uvažujme k oprácií INSERT, každá z týchto operácií vloží do zoznamu rovnaké prirodzené číslo z . Po poslednej z týchto operácií bude mať zoznam dĺžku k .

Cena týchto operácií dohromady je k .

Po týchto operáciách nasleduje l operácií MIN-ALL. Všetky čísla v zozname sú rovnaké, teda všetky čísla v ňom sú minimálne. Znamená to, že pri žiadnom z volaní operácie MIN-ALL sa dĺžka zoznamu nezmení.

Cena týchto operácií bude

$$l * k = \begin{cases} \frac{n}{2} * \frac{n}{2} = \frac{n^2}{4} & \text{ak } n \text{ je párne} \\ \frac{n-1}{2} * (\frac{n}{2} + 1) = \frac{n^2}{4} + \frac{n}{4} - \frac{1}{2} & \text{ak } n \text{ is nepárne} \end{cases}$$

Z predošlého tvrdenia vyplýva, že špecifikovaná postupnosť operácií bude minimimálne v zložitosťnej triede $O(n)$, teda tvrdenie **neplatí**.

Tvrdenie 2: Ľubovoľná postupnosť n operácií INSERT a MIN-ONE má zložitosť $O(n)$. Príklad riešime pomocou metódy účtov, kredity pre jednotlivé operácie stanovíme nasledovne

Operácia	Cena	Kredit
INSERT	1	2
MIN-ONE	$ S $	1

Platí, že vždy počas výpočtu je veľkosť zoznamu rovná počtu kreditov na účte, teda počet kreditov nikdy nebude menší ako 0. Celkový kredit po vykonaní n operácií bude menší alebo rovný $2n$, teda tvrdenie **platí**.

Tvrdenie 3: Ľubovoľná postupnosť n operácií INSERT a DELETE má zložitosť $O(n)$.

Uvažujme prirodzené čísla n, k a l , pre ktoré platí $n=k+l$ (n vyjadruje počet operácií). Hodnotu čísel k a l stanovíme rovnako, ako pri tvrdení 1.

Uvažujme k oprácií INSERT, každá z týchto operácií vloží do zoznamu rovnaké prirodzené číslo z . Po poslednej z týchto operácií bude mať zoznam dĺžku k .

Cena týchto operácií dohromady je k .

Po týchto operáciách nasleduje l operácií DELETE(y), pričom platí, že $y \neq z$. To má za dôsledok, že po žiadnej z týchto operácií sa dĺžka zoznamu nezmení. Cena týchto operácií bude rovnaká, ako v tvrdení 1. Tvrdenie preto **neplatí**.

Tvrdenie 4: Ľubovoľná postupnosť n operácií INSERT a DELETE taká, že pri každom volaní sa operácia DELETE volá s iným parametrom i , má zložitost' $O(n)$.

Uvažujme prirodzené čísla n, k a l , pre ktoré platí $n = k + l$ (n vyjadruje počet operácií). Hodnotu čísel k a l stanovíme rovnako, ako pri tvrdení 1.

Uvažujme k oprácií INSERT, každá z týchto operácií vloží do zoznamu rovnaké prirodzené číslo z . Po poslednej z týchto operácií bude mať zoznam dĺžku k .

Cena týchto operácií dohromady je k .

Špecifikujeme množinu M o veľkosti l , v ktorej sa nachádzajú prirodzené čísla odlišné od z . Vykonáme l operácií DELETE, pričom pri každej jej volaní predložíme ako parameter iný prvok z množiny M . To bude mať za následok, že veľkosť zoznamu sa nezmení. Cena týchto operácií bude rovnaká, ako v tvrdení 1. Tvrdenie preto **neplatí**.

PRÍKLAD 5

LITERATÚRA

- [1] BLUM, Manuel, Robert W. FLOYD, Vaughan PRATT, Ronald L. RIVEST a Robert E. TARJAN. Time bounds for selection. Journal of Computer and System Sciences. 1973, vol. 7, issue 4, s. 448-461. DOI: 10.1016/S0022-0000(73)80033-9. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0022000073800339>
- [2] <http://moonflare.com/code/select/select.pdf>