Problém výberu i-teho prvku

Úloha: Dané je pole n prvkov $A[1 \ldots n]$, číslo i
Nájdi i-ty najmenší prvok v poli.

Príklad: 7,4,8,2,4 $\quad i = 3$

Možné jednoduché riešenia:

- Opakovane vyhľadaj najmenšie číslo. $O(in)$
  (čo ak napríklad hľadáme medián? $i = \lceil n/2 \rceil$)
- Zotrieď pole a vyber i-ty prvok. $O(n \log n)$

Dnes: snaha o lineárne riešenia

```
function SELECT(A,i)
  // find i-th element in array A
  p:=choose_pivot(A);
  //--- partition A into LESS, EQUAL, MORE
  create new arrays LESS, EQUAL, MORE;
  for i:=1 to size(A) do
    if A[i]<p then add A[i] to LESS;
    if A[i]=p then add A[i] to EQUAL;
    if A[i]>p then add A[i] to MORE;
  //--- decide, what case to pursue
  if size(LESS)>=i then
    return SELECT(LESS,i);
  else if size(LESS)+size(EQUAL)>=i then
    return p;
  else
    return SELECT(MORE,i-size(LESS)-size(EQUAL));
```

Akým spôsobom si vyberáme pivot?

(ak by sme vyberali ľubovoľný prvok a máme smolu, tak $O(n^2)$)

1. Rozdeľ pole $A[1 \ldots n]$ na $n/5$ skupín po 5 prvkoch
2. Z každej skupiny vyber tretí najmenší $\Rightarrow$ pole MEDIANS
3. Z poľa MEDIANS vyber medián (rekurzívne zavolaj SELECT)
4. Výsledný prvok vezmeme ako pivot p

aspoň 1/4 prvkov z poľa A je $\leq p$; aspoň 1/4 prvkov je $\geq p$

```
function SELECT(A,i)
  // find i-th element in array A
  p:=choose_pivot(A);
  //--- partition A into LESS, EQUAL, MORE
  create new arrays LESS, EQUAL, MORE;
  for i:=1 to size(A) do
    if A[i]<p then add A[i] to LESS;
    if A[i]=p then add A[i] to EQUAL;
    if A[i]>p then add A[i] to MORE;
  //--- decide, what case to pursue
  if size(LESS)>=i then
    return SELECT(LESS,i);
  else if size(LESS)+size(EQUAL)>=i then
    return p;
  else
    return SELECT(MORE,i-size(LESS)-size(EQUAL));
```

aspoň 1/4 prvkov z poľa A je $\leq p$; aspoň 1/4 prvkov je $\geq p$

- V každom kroku rozdelíme na LESS, EQUAL, MORE
- Ak sme nenašli správny prvok, ďalej vyhľadávame, už len v LESS alebo v MORE
- V každom prípade nám na ďalšie kolo ostane max. 3/4 prvkov lebo |LESS + EQUAL|$\geq 1/4n$
  |MORE + EQUAL| $\geq 1/4n$

Celkový čas behu:

$$
\begin{aligned}
T(n) &= \Theta(n)(1 + 3/4 + (3/4)^2 + (3/4)^3 + \ldots) \\
&= \Theta(n)\tfrac{1}{1-3/4} = \Theta(4n) = \Theta(n)
\end{aligned}
$$

Na čo sme zabudli?

Na čo sme zabudli?

Selekcia pivota:

1. Rozdeľ pole $A[1 \ldots n]$ na $n/5$ skupín po 5 prvkoch
2. Z každej skupiny vyber tretí najmenší $\Rightarrow$ pole MEDIANS
3. Z poľa MEDIANS vyber medián (rekurzívne zavolaj SELECT)
4. Výsledný prvok vezmeme ako pivot p

Správna rekurencia:

$$T(n) = T(\lceil n/5 \rceil) + T(\lfloor 3/4n \rfloor) + \Theta(n)$$

Ľahko ukážeme indukciou, že aj v tomto prípade $T(n) \leq cn$ a teda časová zložitosť je $T(n) = \Theta(n)$.

Zhrnutie deterministického riešenia problému výberu

- Komplikovaný algoritmus na výber pivota
- Garantuje, že v každom kole sa zbavíme aspoň 1/4 prvkov
- Výsledný algoritmu má zložitosť $\Theta(n)$

Pravdepodobnostný algoritmus (randomized algorithm)

- Idea: Vyber pivot ako **náhodný prvok poľa**
- Intuícia: Obvykle to bude fungovať celkom dobre
- V najhoršom prípade: vždy zhodou okolností zvolíme minimum alebo maximum

Časová zložitosť deterministického algoritmu

Časová zložitosť algoritmu A je **funkciou veľkosti vstupu**, pričom $T_A(n)$ je najväčší čas potrebný na vyriešenie vstupu o veľkosti n.

$T_A(n) = \max\{T_A(x) \mid |x| = n\}$

($T_A(x)$ je čas, ktorý algoritmus A potrebuje na vyriešenie vstupu x.)

Očakávaná časová zložitosť pravdepodobnostného algoritmu

Časová zložitosť algoritmu A je **funkciou veľkosti vstupu**, pričom $T_A(n)$ je najväčší čas potrebný na vyriešenie vstupu o veľkosti n.

$T_A(n) = \max\{E_R[T_A(x, R)] \mid |x| = n\}$

($T_A(x, R)$ je čas, ktorý algoritmus A potrebuje na vyriešenie vstupu x za predpokladu, že postupnosť príslušných náhodných rozhodnutí je R.)

Počítame časovú zložitosť algoritmu s náhodnou voľbou pivota

$T_A(n) = \Theta(n) + E[T_A(n')] \leq \Theta(n) + \sum_{i=1}^{n} \frac{1}{n} T_A(i-1)$

$T_A(n) \leq \Theta(n) + \frac{1}{2} T_A(3/4n) + \frac{1}{2} T_A(n-1)$

Riešime rekurenciu...

$T_A(n) \leq \Theta(n) + \frac{1}{2}T_A(3/4n) + \frac{1}{2}T_A(n-1)$

Indukciou dokážeme, že pre vhodné $c > 0$ platí $T_A(n) \leq cn$.

Báza: Pre dostatočne malé n zjavne platí, ak c je dostatočne veľké.

Indukčný krok: Prepokladajme, že platí $T_A(n') \leq cn'$ pre $n' < n$.

$$\begin{aligned} T_A(n) &\leq c'n + \frac{1}{2}\frac{3}{4}nc + \frac{1}{2}(n-1)c \\ &\leq c'n + \frac{3}{8}nc + \frac{4}{8}nc \\ &= c'n + \frac{7}{8}nc \\ &\leq cn \end{aligned}$$

(za predpokladu, že $c' \leq 1/8c$)

Čiže, $T_A(n) \leq cn = O(n)$.

Dokázali sme, že **očakávaná časová zložitosť** pravdepodobnostného algoritmu je $O(n)$.

Zhrnutie

- Problém výberu i-teho nejmenšieho prvku
  (napríklad hľadanie mediánu)
- (Veľmi zložitý) lineárny deterministický algoritmus, ktorý nie je veľmi praktický
- Veľmi praktický pravdepodobnostný algoritmus
  Čas jednotlivých behov sa môže líšiť podľa toho, ako nám "padnú karty" (tzv. **Las Vegas** algoritmus)
- Pojem očakávanej časovej zložitosti
- Lineárna očakávaná časová zložitosť pravdepodobnostného algoritmu pre výber