

Nalezení k-tého nejmenšího prvků posloupnosti $x_1 \dots x_n$

- 1) Vyberu pivot $\in x_1 \dots x_n$
- 2) Rozdělím vstup na L, S, P
 $\leq p, = p, > p$
- 3) Pokud $k \leq |L|$: rekureze na L, k
 Pokud $k \leq |L| + |S|$: vrátíme pivot
 Jinak rekureze na $P, k - |L| - |S|$

} Rychlost závisí na faktoru zmenšení

Výběr pivotu

- maximum $\Rightarrow \Theta(n^2)$ zatožní všechny j+1
- medián $\Rightarrow \Theta(n)$ celková
- pseudomedián $\Rightarrow \Theta(n)$
- náhodně $\Rightarrow E = O(n)$

$\hookrightarrow$ přestože v nejhorším $O(n^2)$

Select ($x_1 \dots x_n; k$):

- 1) Rozdělíme $x_1 \dots x_n$ na pětky $p_1 \dots p_t$ ($t = \lceil \frac{n}{5} \rceil$)
- 2) Najdeme mediány pětic: $\forall i: m_i \leftarrow \text{medián}(p_i)$
- 3) Najdeme medián mediánů pětic:
 $: \text{Select}(m_1 \dots m_t, \lceil \frac{t}{2} \rceil)$

$\hookrightarrow$ pivot z předchozího algoritmu

Věta: Select má časovou složitost $\Theta(n)$

Důk:

$\frac{n}{5}$ pětic $\rightarrow$ celkem $\frac{3}{10}n$ prvků

$\geq p$ $\rightarrow$ mediány pětic

$\leq p$ $\rightarrow$ $\frac{n}{5}$ pětic

Alespoň jednou takovou část musím zabít.

Ukážeme $T(n) = cn$

$$cn = \frac{1}{5}cn + \frac{7}{10}cn + n$$

$$\frac{1}{10}cn = n$$

$$\boxed{c = 10} \Rightarrow T(n) = cn$$

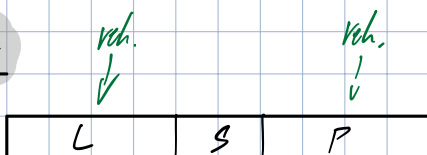
$\hookrightarrow$ Tedy v nejhorším případě má lineární složitost

$\hookrightarrow$ Tedy určitě alespoň $\frac{3}{10}$ prvků zabijeme v každém kroku
 $\Rightarrow$ zbývá $\frac{7}{10}$ prvků nejvýše

$$T(n) = T(n/5) + T(\frac{7}{10}n) + \cancel{\Theta(n)}^n$$

$$T(n) = \Theta(n)$$

Quick Sort



0. Pokud $n \leq 1$: vrátíme vstup
1. Zvolíme pivot p .
2. Rozdělíme $x_1 \dots x_n$ podle L, S, P
3. $L \leftarrow \text{quicksort}(L)$
 $R \leftarrow \text{quicksort}(R)$ → složitost
4. Vrátíme $L \parallel S \parallel P$.

Složitost:

- pokud p bude median.

$$T(n) = 2T(n/2) + \Theta(n)$$

↓

$$T(n) = \Theta(n \log n)$$

→ už umíme najít pivot v čase $O(n)$

- pokud p min/max:

$$T(n) = T(n-1) + \underbrace{T(0)}_{\Theta(n)} + \Theta(n)$$

↓

$$T(n) = \Theta(n^2)$$

Věta: QuickSort s náhodnou volbou pivotu má časovou složitost se střední hodnotou $\Theta(n \log n)$

Důkaz #1:

Porovnání, kolikrát se liší při zúčastnění porovnání.

- porovnání účtujeme pro každý, který nebyl pivotem (ten bude porovnán fakt...)

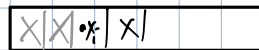
Kolik porovnání jsme navštívili x_i ? $\rightarrow P_i$

$$\text{čas} = \Theta(\# \text{porovnání})$$

$$\# \text{porovnání} = \sum P_i$$

$$E[\# \text{porovnání}] = \sum E[P_i]$$

Sledujeme velikost podproblémů, ve kterých je x_i .



- pokud použijeme pseudomedian jako pivot, tak v každém problému vyhodíme alespoň $\frac{1}{5}$ prvků

Dělíme na fáze: každá volba $p_i = \text{pseudomedian}$

- ① Fáze zanechá n má $\frac{2}{5}n$: $\# \text{fází} \in O(\log n)$
- ② $E[\text{velikost ve fázi}] \leq c$

Takže jedna fáze trvá $O(\log n)$, celkem může být až n porovnání, tedy celkem $O(n \log n)$

Důkaz #2: $y_1 \dots y_n$ setříděné pořadí prvků (výstup)

$$C_{ij} : 1 \leq i < j \leq n$$

$$C_{ij} \in \{0, 1\}$$

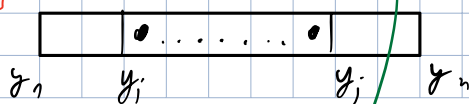
1 pokud jsme porovnali y_i s y_j (stačí se nejvýše jednou)

Při porovnání je jeden z dvou pivotů a ten nepostupuje do rekurence

$$\text{Celkový \# porovnání } C = \sum_{i,j} C_{ij}$$

$$E[C] = \sum_{i,j} E[C_{ij}]$$

$$E[C_{ij}] = P[y_i, y_j \text{ bylo porovnání}]$$



→ y_i, y_j se účastní porovnání, pokud jeden z nich je pivot a pokud jsou ve stejném intervalu.

Všechny dokazovat

$$E[C] = \sum_{1 \leq i < j \leq n} \frac{2}{j-i+1} =$$

$$= \sum_{j=2}^n \sum_{i=1}^{j-1} \frac{2}{j-i+1}$$

délka intervalu

$$\sum_{1 \leq i < j \leq n} \frac{2}{j-i+1} \leq \sum_{d=1}^n \frac{2}{d} \cdot (n-d+1) \leq 2n \sum_{d=1}^n \frac{1}{d} \in \Theta(n \log n)$$

harmonický součet - roste logaritmičtě rychle

- tedy žádný prvek mezi y_i, y_j ještě nemohl být pivotem, protože y_i a y_j by se potkaly až při finálním dělení.

- tedy y_i nebo y_j se z prvků $y_i - y_j$ stalo pivotem jako první.

→ harmonický řada

Lemma: $\frac{1}{x} \ln n \leq H_n \leq \ln n + 1$



$$I_n = \int_1^n \frac{1}{x} dx = [\ln x]_1^n = \ln n$$

I_5
tabulka se
vkládá do
oblasti, co
integrujeme

$$H_n = I_n + \text{red area} \rightarrow H_n \leq I_n + 1 = \ln n + 1$$

$$H_n = I_n + \text{red area} \rightarrow H_n \geq I_n + \frac{1}{n} \geq I_n = \ln n$$

- tabulka jsme to zvrhli musí dá příslušy

Dynamická programování

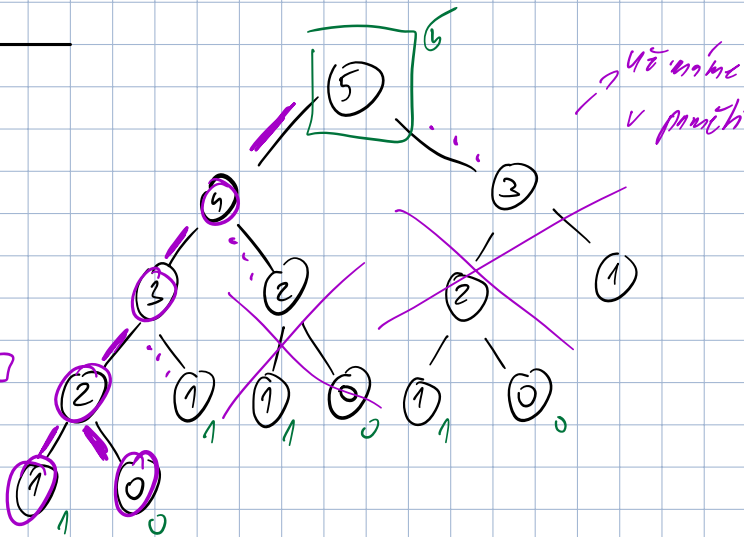
Výpočet Fib. čísel:

minimální
 $P[0] = 0$

$F(n)$: pokud $P[n] \neq 0$: vrátíme $P[n]$

1) Pokud $n \leq 1$: vrátíme n

2) Jinak vrátíme $F(n-1) + F(n-2)$



$F(n)$ je součet listů v podstromu

$$F_n \geq 2^{n/2} \rightarrow \text{exponenciální růst!}$$

→ jiná možnost: Vycházet $P[0] - P[n] \geq n/2$ známe

$\hookrightarrow$ více $P[i]$ potřebuje znát $P[i-1] + P[i-2]$

- počítáme spoustu věcí znovu, takže si to cachejeme.

- Uložíme $F(i)$ pro $0 \leq i \leq n$ počítáme pouze jednou a trvá to $O(n)$,
tedy celkem $O(n)$

Postup:

- začneme rekursivním alg, který je zpravidla velmi pomalý
- 😊 počítáme modnat to stejné
- paměť na známé výsledky (cache)
- L = polg složitost
- rekursi vyvoláme za vyplnění cache v polynomiálním čase

Hledáme nejdelší vybranou podsekvenci v $x_1 \dots x_n$ (kde $x_0 = -\infty, x_{n+1} = +\infty$)

$NRP(i)$: $\rightarrow$ délka NRP vybrané z $x_i \dots x_{n+1}$
začínající x_i

1. $d \leftarrow 1$ (zabírá nejvyšší délku)

2. Pro $j = i+1 \dots n+1$

3. Pokud $x_i < x_j$:

4. $d \leftarrow \max(NRP(j)+1, d)$

5. Vraťme d .

① Opět exponenciální postupnost

② $i \in \{0 \dots n+1\}$

$\rightarrow$ volání se opakuje

③ Přidání cache

$\rightarrow$ pro zadání $O(n)$ - máť
potřebná počítat v čase $O(n)$

④ Vyplňujeme tabulku cyklem

1. $P[n+1] \leftarrow 1$

2. Pro $i = n \dots 0$ pozpátku:

3. $d \leftarrow 1$

4. Pro $j = i+1 \dots n+1$

5. Pokud $x_i < x_j$

6. $d \leftarrow \max(P[j]+1, d)$

7. $P[i] \leftarrow d$

8. Vraťme $P[0]$

- nyní běží v $O(n^2)$