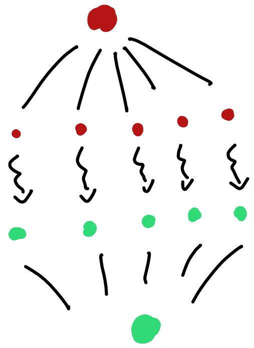
2. Deli in vladaj



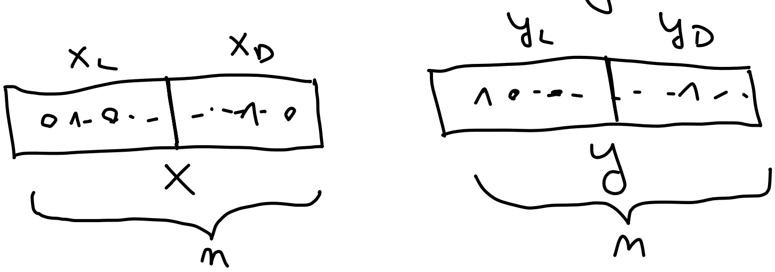
razbijemo na manjše podprobleme — DELI

IN — rekurzivno rešimo

združimo v rešitev prvotnega problema — VLADAJ

- iskanje z bisekcijo
- algoritmi za urejanje
- hitro potenciranje

2.1. Karatsubov algoritem



$$x = 2^{n/2} \cdot x_L + x_D$$

$$y = 2^{n/2} \cdot y_L + y_D$$

$$x \cdot y = 2^n \cdot x_L y_L + 2^{n/2} (x_L y_D + x_D y_L) + x_D y_D$$

Karatsuba
*1937 Grozni, ZSSR
† 2008 Moskva, Rusija

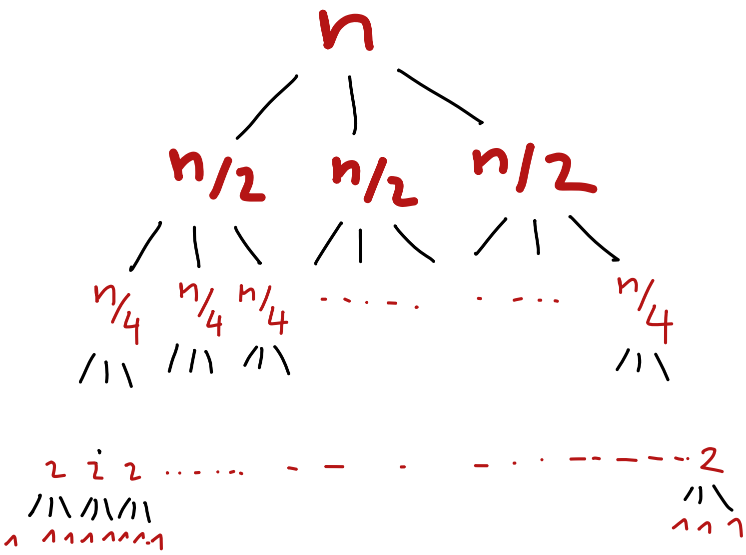
$$x_L y_D + x_D y_L = (x_L + x_D) \cdot (y_L + y_D) - x_L y_L - x_D y_D$$

(1) ✓

(2) $T(n) = 3 \cdot T(n/2) + O(n)$

$$= O(n^{\log_2 3}) \approx O(n^{1,58})$$

(3) Da se še hitreje. $O(n \log n)$ .... FFT ... PSA2



$$\left.\begin{array}{l} O(n) \\ 3\cdot O\left(\frac{n}{2}\right) \\ 9\cdot O\left(\frac{n}{4}\right) \\ 3^k \cdot O\left(\frac{n}{2^k}\right) \end{array}\right\} \log n$$

$$T(n) = \sum_{k=0}^{\log n} 3^k \cdot O\left(\frac{n}{2^k}\right) = O(n) \sum_{k=0}^{\log n} \left(\frac{3}{2}\right)^k$$

$$1 + x + \dots + x^n = \frac{x^{n+1} - 1}{x - 1}$$

$$= O(n) \frac{\left(\frac{3}{2}\right)^{\log n + 1} - 1}{\frac{3}{2} - 1}$$

$$a^{\log b} = 2^{\log a \log b} = b^{\log a}$$

$$= O(n)\, 2\left(\left(\frac{3}{2}\right) \cdot \frac{3^{\log n}}{2^{\log n}} - 1\right)$$

$$= O(\cancel{n})\, \cancel{2}\left(\frac{3}{\cancel{2}} \cdot \frac{3^{\log n} - 2n}{\cancel{n}}\right)$$

$$= O\left(3 \cdot 3^{\log n} - 2n\right)$$

$$= O\left(\cancel{3} \cdot n^{\log 3} - \cancel{2n}\right)$$

$$= O\left(n^{\log 3}\right)$$

2.2. Krovni izrek

Izrek Če velja $T(n) = a \cdot T(n/b) + O(n^d)$,

potem velja

$$T(n) = \begin{cases} O(n^d) & a < b^d & \log_b a < d \\ O(n^d \log n) & a = b^d & \log_b a = d \\ O(n^{\log_b a}) & a > b^d & \log_b a > d \end{cases}$$

Primeri

- iskanje z bisekcijo

  $a = 1 \quad b = 2 \quad d = 0 \qquad b^d = 1 = a$

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

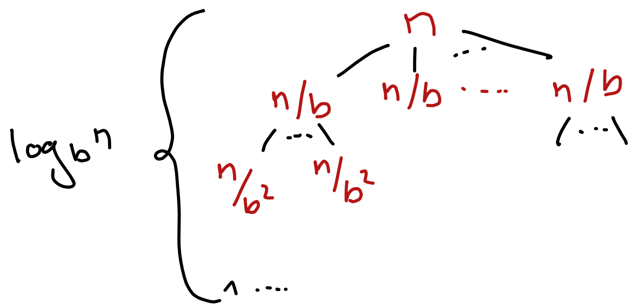
- Karatsuba

  $b^d = 2 < 3 = a$

  $a = 3 \quad b = 2 \quad d = 1$

  $T(n) = O(n^{\log_2 3})$

Dokaz



$$O(n^d)$$
$$a \cdot O\left(\left(\frac{n}{b}\right)^d\right)$$
$$a^2 \cdot O\left(\left(\frac{n}{b^2}\right)^d\right)$$
$$\vdots$$

$$O(1+c+c^2+\dots+c^m) = \begin{cases} O(1) & c<1 \\ O(m) & c=1 \\ O(c^m) & c>1 \end{cases}$$

$$T(n) = \sum_{k=0}^{\log_b n} a^k \cdot O\left(\left(\frac{n}{b^k}\right)^d\right)$$

JohnvonNeumann-LosAlamos.gif

$$= \sum_{k=0}^{\log_b n} \left(\frac{a}{b^d}\right)^k \cdot O(n^d)$$

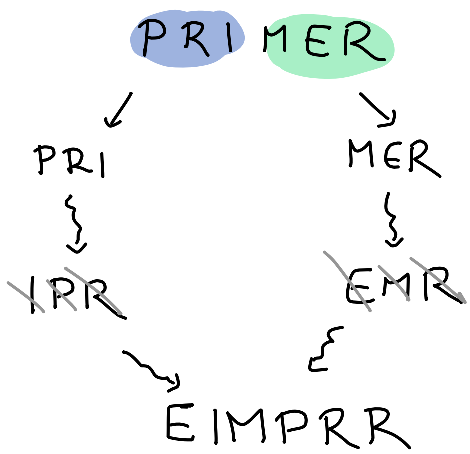
$$= O(n^d) \cdot \sum_{k=0}^{\log_b n} \left(\frac{a}{b^d}\right)^k$$

$$= \begin{cases} O(n^d) \cdot O(1) & a<b^d \\ O(n^d) \cdot O(\log_b n) & a=b^d \\ O(n^d) \cdot O\left(\left(\frac{a}{b^d}\right)^{\log_b n}\right) & a>b^d \end{cases}$$

$$= \begin{cases} O(n^d) & a<b^d \\ O(n^d \log_b n) & a=b^d \\ O(n^{\log_b a}) & a>b^d \end{cases}$$

$\blacksquare$

2.3. Urejanje z zlivanjem



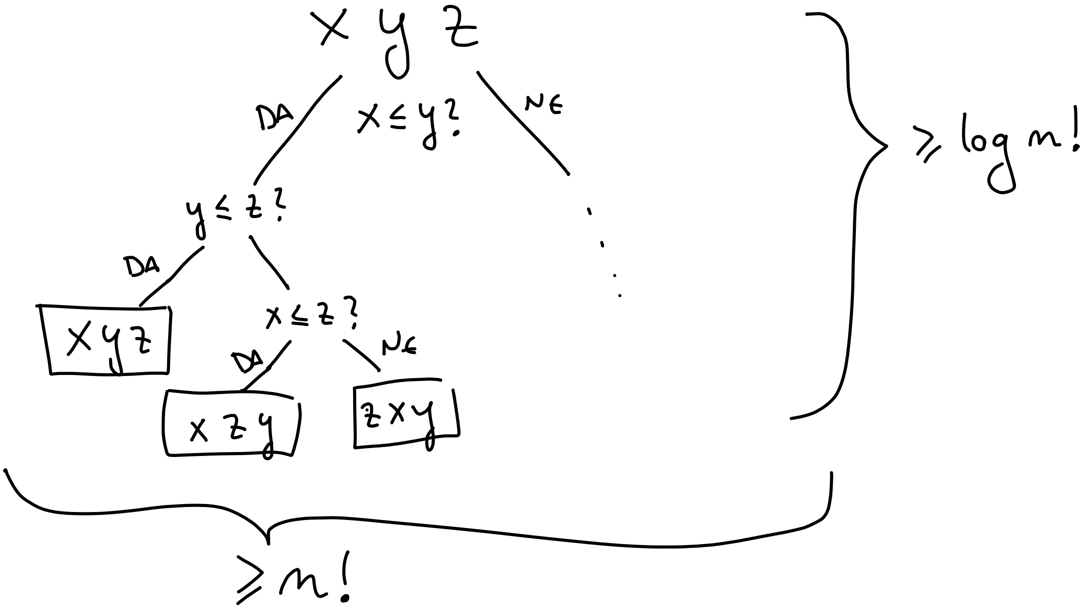
1. ✓

2. $T(n) = 2 \cdot T(n/2) + O(n) = O(n \cdot \log n)$

$a=2 \quad b=2 \quad d=1$

$b^d = 2 = a$

3. $T(n) = \Omega(n \cdot \log n)$



$$m! = \underbrace{m\cdot(m-1)\cdots}_{\text{vsak} \geq m/2}\underbrace{3\cdot 2\cdot 1}_{\text{vsak} \geq 1} \geq (m/2)^{m/2}$$

$$\log(m!) \geq \log\left((m/2)^{m/2}\right) = \frac{m}{2}\cdot\left(\log(m)-\log(2)\right) = \Theta(m\log m)$$

Von Neumann
*1903 Budimpesta, A-O
† 1957 Washington D.C., ZDA

2.4. Iskanje mediane

↳ po vrsti srednji element

Lahko uredimo seznam in vzamemo srednji element.

① ✓

② $O(n \log n)$

③ ja, se da

Hoare

* 1934 Kolombo, Cejlon

Hitro izbiranje (quick-select)

vhod seznam a dolžine n, $k \in \{1, \dots, n\}$

izhod po vrsti k-ti element v a

def quickSelect(a, k):

    če $|a| = 1$:

        vrni a_1

    sicer:

        izberi $p \in a$ (pivot)

        $a_m, a_e, a_v =$ a razdeljen na elemente, ki so $<p, =p, >p$

        če $k \leq |a_m|$:

            vrni quickselect(a_m, k)

        če $k \leq |a_m| + |a_e|$:

            vrni p

        sicer:

            vrni quickselect$(a_v, k - |a_m| - |a_e|)$

2 13 7 -6 5 0 2

a_m a_e a_v

-6 0 2 2 13 7 5

①

②

$a=1 \quad b=2 \quad d=1 \qquad b^d = 2 > a$

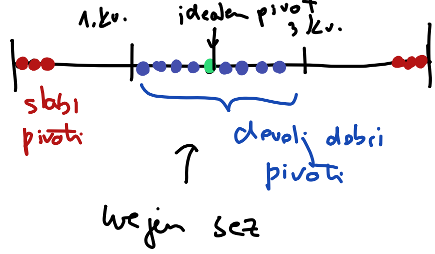
$$T(n) = \begin{cases} O(n) + T(n/2) = O(n) \\ O(n) + T(n-1) = O(n^2) \end{cases}$$

↳ na to upamo in to se v povprečju tudi zgodi

↳ v najslabšem primeru in to se zgodi, če je sez. urejen in za p vzamemo prvi element

③ Da se boljše, radi bi se izognili $O(n^2)$

Pravimo, da je pivot dober, če leži med 1. in 3. kvartilom.



ps = a

ponavljaj:

izberi naključen $p \in ps$

a_m, a_p, a_v = elementi $<p, =p, >p$

če p dober pivot: $|a_m| \leq \frac{3}{4} \cdot |a| \wedge |a_v| \leq \frac{3}{4} |a|$

vrni p

ps = am oz. av

Koliko obhodov pričakujemo?

$$E(o) = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot (1 + E(o)) \Rightarrow E(o) = 2$$

Skupen čas, ki ga pričakujemo za tako pivotiranje je $2 \cdot O(n) = O(n)$

Čas za iskanje k-tega elementa je:

$$T(n) = O(n) + T\left(\frac{3}{4} n\right) = O(n)$$

$d = 1 \quad a = 1 \quad b = 4/3 \qquad b^d = 4/3 \geq a$

Na podoben način bi lahko prilagodili tudi hitro urejanje.

(3) k-tega elementa hitreje kot v $O(n)$ ne moremo najti.

Recimo, da imamo algoritem, ki za

$a = [1, 1, \ldots, 1]$

$k = 1$

rezultat najde hitreje kot $O(n)$. Potem ni mogel pogledati vseh elementov, konkretno ni pogledal a_j

Zdaj isti algoritem poženemo na

$$a' = [1, 1, \ldots 1, \underset{\uparrow j}{0}, 1 \ldots, 1]$$

$k = 1$

Ker algoritem ne pogleda a'_j, se odloči enako kot prej za rezultat 1, ki je napačen.

2.5. Strassenovo množenje

Množenje matrik $n \times n$

- običajno: $O(n^3)$
- Strassenov alg. $T(n) = 7 \cdot T(n/2) + O(n^2) = O(n^{\log_2 7})$

Strassen

* 1936

Düsseldorf,

Nemčija

2.6. FFT (Hitra fourierova transformacija)

PSA 2