Algoritmi in podatkovne strukture 1

Visokošolski strokovni študij Računalništvo in informatika

Deli in vladaj

Jurij Mihelič, UniLj, FRI

Deli in vladaj

Gaj Julij Cezar
100 pr. n. št. – 44 pr. n. št.

- Divide et impera (*divide & conquer*)
  - Deli:
    - problem **delimo** na
    - **manjše** probleme
    - dokler ne dobimo
    - ***obvladljivega*** problema
  - Vladaj
    - majhne probleme
    - enostavno oz. trivialno rešimo

Deli in vladaj

- Metoda snovanja algoritmov
  - rekurzivni algoritem
- Koraki
  - delitev naloge:
    - na eno ali več **manjših** nalog
  - *reši manjše naloge:*
    - uporaba rekurzije
  - združevanje rešitev:
    - iz rešitev manjših nalog sestavimo rešitev osnovne naloge

Dvojiško iskanje

- Ideja algoritma
  - tabelo delimo na **dve** polovici
  - rekurzija gre le v **eno** polovico
  - vladaj: tabela velikosti 1
  - zahtevnost delitve $O(1)$ in sestavljanja $O(1)$
- Rekurenčna enačba

Dvojiško iskanje (rekurzivno)

```
fun binarySearch(a, left, right, key) is
    if right > left then return -1
    mid = left + (right – left) / 2
    if (key < a[mid]) then
        return binarySearch(a, left, mid – 1)
    if (k > a[mid]) then
        return binarySearch(a, mid + 1, right)
    return mid
```

Urejanje z zlivanjem

- Ideja algoritma
  - tabelo delimo na **dve** polovici
  - rekurzija gre v **obe** polovici
  - vladaj: tabela velikosti 1
  - zahtevnost delitve $O(1)$ in sestavljanja $\underline{O}(n)$

»Master theorem«

- Rekurenčna enačba

$$T(n)=a\cdot T\left(\frac{n}{b}\right)+c\,n^{d}$$

a *– število podnalog (podproblemov)*

b *– faktor delitve naloge*

c – konstanta iz asimptotične notacije

d *– red velikost zahtevnosti delitve in združevanja.*

»Master theorem«

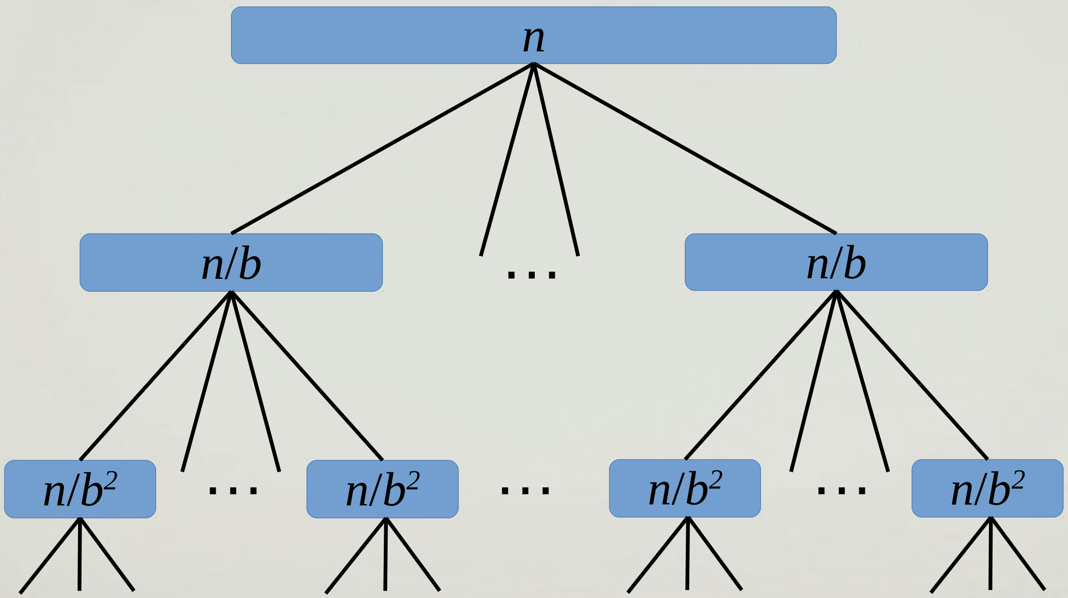
- Rekurenčna enačba in rešitev

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + c\,n^d$$

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

»Master theorem«

- Intuicija $$T(n) = a \cdot T\left(\frac{n}{b}\right) + c\,n^d$$



»Master theorem«

$$\left(\frac{a}{b^d}\right)^i n^d$$

- Sile zla a vs sile dobrega b^d
  - $a < b^d \Rightarrow \Theta(n^d)$
    - večja globina ► manj dela ► največ dela v korenu
  - $a = b^d \Rightarrow \Theta(n^d \lg n)$
    - na vsaki globini je enako dela
  - $a > b^d \Rightarrow \Theta(n^{\log_b a})$
    - večja globina ► več dela ► največ dela v listih

»Master theorem«

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

- Primeri
  - dvojiško iskanje: (1,2,0)
  - kopica – dvigovanje: (1,2,0)
  - kopica – ugrezanje: (1,3/2,0)
  - urejanje z zlivanjem: (2,2,1)
  - štetje inverzij: (2,2,1)
  - hitro urejanje (najboljši primer): (2,2,1)
  - množenje celih števil – D&V: (4,2,1)
  - množenje celih števil – Karatsuba: (3,2,1)
  - množenje matrik – D&V: (8,2,2)
  - množenje matrik – Strassen: (7,2,2)

Množenje velikih celih števil

- Delitev števil
  - n-bitna števila razdelimo na pol
  - na dva $n/2$ bitna dela
    - prva polovica
    - druga polovica

n bitov

$n/2$ bitov | $n/2$ bitov

$$a = a_1 \cdot 2^{n/2} + a_0$$

$$b = b_1 \cdot 2^{n/2} + b_0$$

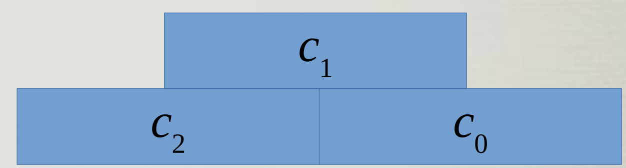
Množenje velikih celih števil

- Direktno D&V množenje
  - n-bitna števila razdelimo na pol

    $$a = a_1 \cdot 2^{n/2} + a_0 \text{ in } b = b_1 \cdot 2^{n/2} + b_0$$

  - produkt

    $$\begin{aligned} a \cdot b &= (a_1 \cdot 2^{n/2} + a_0) \cdot (b_1 \cdot 2^{n/2} + b_0) \\ &= a_1 b_1 \cdot 2^{n} + (a_1 b_0 + a_0 b_1) \cdot 2^{n/2} + a_0 b_0 \\ &= c_2 \cdot 2^{n} + c_1 \cdot 2^{n/2} + c_0 \end{aligned}$$

    $$\begin{aligned} c_2 &= a_1 \cdot b_1 \\ c_1 &= a_1 b_0 + a_0 b_1 \\ c_0 &= a_0 \cdot b_0 \end{aligned}$$

  - zahtevnost
    - $O(n^2)$

Množenje velikih celih števil

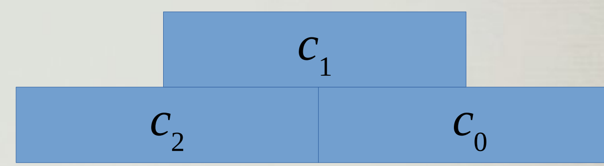
A. A. Karacuba, 1937 – 2008

- Karatsubov algoritem
  - n-bitna števila razdelimo na pol

$$a = a_1 \cdot 2^{n/2} + a_0 \quad \text{in} \quad b = b_1 \cdot 2^{n/2} + b_0$$

  - produkt

$$\begin{aligned} a \cdot b &= (a_1 \cdot 2^{n/2} + a_0) \cdot (b_1 \cdot 2^{n/2} + b_0) \\ &= a_1 b_1 \cdot 2^n + (a_1 b_0 + a_0 b_1) \cdot 2^{n/2} + a_0 b_0 \\ &= c_2 \cdot 2^n + c_1 \cdot 2^{n/2} + c_0 \end{aligned}$$



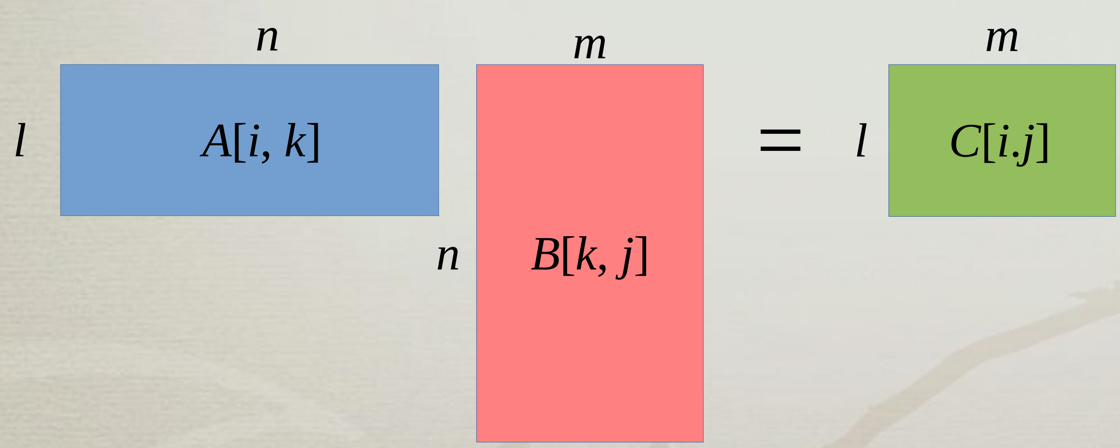
  - Gaussov / Karatsubov trik

$$\begin{aligned} c_2 &= a_1 \cdot b_1 \\ c_0 &= a_0 \cdot b_0 \\ c_1 &= (a_1 + a_0) \cdot (b_1 + b_0) - c_2 - c_0 \end{aligned}$$

  - zahtevnost
    - $T(n) = O(n^{\lg 3}) = O(n^{1.585})$

Množenje matrik

- Problem
  - dani sta dve matriki A in B
  - iščemo njun produkt $C = AB$
- Klasični algoritem
  - tri zanke for
  - zahtevnost $O(n^3)$



Množenje matrik

- Delitev matrik na manjše

$$A \rightarrow \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix}$$

- Vsota
  - $C_{00} = A_{00} + B_{00}$
  - $C_{01} = A_{01} + B_{01}$
  - $C_{10} = A_{10} + B_{10}$
  - $C_{11} = A_{11} + B_{11}$

$$\begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} + \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{bmatrix}$$

- Produkt
  - $C_{00} = A_{00}B_{00} + A_{01}B_{10}$
  - $C_{01} = A_{00}B_{01} + A_{01}B_{11}$
  - $C_{10} = A_{10}B_{00} + A_{11}B_{10}$
  - $C_{11} = A_{10}B_{01} + A_{11}B_{11}$

$$\begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{bmatrix}$$

Množenje matrik

- Direktno D&V množenje

$$\begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{bmatrix}$$

  - Produkti
    - $C_{00} = A_{00}B_{00} + A_{01}B_{10}$
    - $C_{01} = A_{00}B_{01} + A_{01}B_{11}$
    - $C_{10} = A_{10}B_{00} + A_{11}B_{10}$
    - $C_{11} = A_{10}B_{01} + A_{11}B_{11}$
  - Zahtevnost
    - $O(n^3)$

Množenje matrik

V. Strassen, 1936

- Strassenov algoritem

$$\begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{bmatrix}$$

  - Produkti
    - $C_{00} = M_1 + M_4 - M_5 + M_7$
    - $C_{01} = M_3 + M_5$
    - $C_{10} = M_2 + M_4$
    - $C_{11} = M_1 + M_3 - M_2 + M_6$
  - Zahtevnost
    - $O(n^{2.808})$

$M_1 = (A_{00} + A_{11})\,(B_{00} + B_{11})$

$M_2 = (A_{10} + A_{11})\,B_{00}$

$M_3 = A_{00}\,(B_{01} - B_{11})$

$M_4 = A_{11}(B_{10} - B_{00})$

$M_5 = (A_{00} + A_{01})\,B_{11}$

$M_6 = (A_{10} - A_{00})\,(B_{00} + B_{01})$

$M_7 = (A_{01} - A_{11})\,(B_{10} + B_{11})$

Množenje matrik

- Strassenov algoritem v praksi
  - velika konstanta
  - prostorska potratnost
  - naivna metoda je numerično stabilnejša
  - matrike niso vedno velikosti $n = 2^k$
  - ustavljanje rekurzije pri majhnih matrikah
- Asimptotično najhitrejši: $O(n^\alpha)$
  - Coopersmith, Winograd, 1990, $\alpha = 2.37\mathbf{6}$
  - Stothers, 2010, $\alpha = 2.37\mathbf{4}$ ($\alpha = 2.3736$)
  - Williams, 2011, $\alpha = 2.37\mathbf{3}$ ($\alpha = 2.37286\mathbf{42}$)
  - Le Gall, 2014, $\alpha = 2.37286\mathbf{39}$