Algoritmi si Structuri de Date

Bibliografie

1. D.E. Knuth :"Tratat de programarea calculatoarelor", vol. I si III
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest: "Introduction to Algorithms", The MIT Press, 1990
Cormen, Leiserson, Rivest, Stein - CLRS2
3. N. Wirth: "Algorithms + Data Structures = Programs", Prentice Hall Inc., 1976
4. A. V. Aho, J. E. Hopcroft, J. D. Ullman: "Data Structures and Algorithms", Addison-Wesley Publ. Comp., 1983
5. Sedgewick
6. Dasgupta, Papadimitriou, Vazirani

Algoritmi

-

Algoritmi

-

Structuri de Date

-

Algoritmi

-

Structuri de Date

-

"Algorithms + Data Structures = Programs"

- Niklaus Wirth, 1976

Algoritmi

Corectitudine

- se opreste
- relatia 'input/output' dorita

Algoritmi

Corectitudine

- se opreste
- relatia 'input/output' dorita

Performanta

- Masura a costurilor
  - timp (viteza)
  - spatiu

Algoritmi

Corectitudine

- se opreste
- relatia 'input/output' dorita

Performanta

- Masura a costurilor
  - timp (viteza)
  - spatiu

Mai bine?

- Se poate mai bine?

Algoritmi

Corectitudine

- se opreste
- relatia 'input/output' dorita

Performanta

- Masura a costurilor
  - timp (viteza)
  - spatiu

Mai bine?

- Se poate mai bine?

2 Mari Probleme

Sortarea

- **Input** Un vector $(a_1, a_2, \cdots, a_n)$
- **Output** $(a'_1, a'_2, \cdots, a'_n)$ a.i. $a'_1 \leq a'_2 \leq \cdots \leq a'_n$

2 Mari Probleme

Sortarea

- **Input** Un vector $(a_1, a_2, \cdots, a_n)$
- **Output** $(a'_1, a'_2, \cdots, a'_n)$ a.i. $a'_1 \leq a'_2 \leq \cdots \leq a'_n$

Cautarea

- **Input** O 'multime' $\{a_1, a_2, \cdots, a_n\}$ si o 'valoare' x
- **Output** Este $x \in \{a_1, a_2, \cdots, a_n\}$ **?** **Yes** sau **No**

Sortarea

Sortarea prin insertie

Sortarea prin insertie

Sortarea prin insertie (directa)

void InsDir (int A[], int n)
{ //sortare prin insertie directa a vectorului A[1..n]
for ($i = 2$; $i <= n$; $i++$)
{
$x = A[i]$;
//se caută locul valorii x în destinație
$j = i - 1$;
while (($j > 0$) && ($x < A[j]$))
{
$A[j + 1] = A[j]$;
$j - -$;
}
//inserarea lui x la locul lui
$A[j + 1] = x$;
}
}

Sortarea prin insertie

Comparatii

- $C_{min} = 1 + 1 + \cdots + 1 = n - 1$
- $C_{max} = 2 + 3 + \cdots + n = n \cdot (n+1)/2 - 1 = (1/2)n^2 + (1/2)n - 1$
- $C_{mediu} = (1/2)(2+3+\cdots+n) = (1/2)C_{max} = (1/4)n^2 + (1/4)n - 1/2$

Sortarea prin insertie

Comparatii

- $C_{min} = 1 + 1 + \cdots + 1 = n - 1$
- $C_{max} = 2 + 3 + \cdots + n = n \cdot (n + 1)/2 - 1 = (1/2)n^2 + (1/2)n - 1$
- $C_{mediu} = (1/2)(2+3+\cdots+n) = (1/2)C_{max} = (1/4)n^2+(1/4)n-1/2$

- Ce inseamna cazul **mediu**?

Sortarea prin insertie

Comparatii

- $C_{min} = 1 + 1 + \cdots + 1 = n - 1$
- $C_{max} = 2 + 3 + \cdots + n = n \cdot (n+1)/2 - 1 = (1/2)n^2 + (1/2)n - 1$
- $C_{mediu} = (1/2)(2+3+\cdots+n) = (1/2)C_{max} = (1/4)n^2 + (1/4)n - 1/2$

- Ce inseamna cazul **mediu**?

Mutari

- $M_{min} = 2 + 2 + \cdots + 2 = 2(n-1)$
- $M_{max} = 3 + 4 + \cdots + (n+1) = (n+1) \cdot (n+2)/2 - 3 = (1/2)n^2 + (3/2)n - 2$
- $M_{mediu} = C_{mediu} + 2(n-1) = (1/4)n^2 + (9/4)n - 5/2$

Sortarea prin insertie

Comparatii

- $C_{min} = 1 + 1 + \cdots + 1 = n - 1$
- $C_{max} = 2 + 3 + \cdots + n = n \cdot (n+1)/2 - 1 = (1/2)n^2 + (1/2)n - 1$
- $C_{mediu} = (1/2)(2+3+\cdots+n) = (1/2)C_{max} = (1/4)n^2 + (1/4)n - 1/2$

- Ce inseamna cazul **mediu**?

Mutari

- $M_{min} = 2 + 2 + \cdots + 2 = 2(n-1)$
- $M_{max} = 3 + 4 + \cdots + (n+1) = (n+1) \cdot (n+2)/2 - 3 = (1/2)n^2 + (3/2)n - 2$
- $M_{mediu} = C_{mediu} + 2(n-1) = (1/4)n^2 + (9/4)n - 5/2$

- la pasul/iteratia i, avem $M_i = C_i + 1$.

Performanta

Comparatii

- $C_{min} = n - 1$
- $C_{max} = (1/2)n^2 + (1/2)n - 1$
- $C_{mediu} = (1/4)n^2 + (1/4)n - 1/2$

Performanta

Comparatii

- $C_{min} = n - 1$
- $C_{max} = (1/2)n^2 + (1/2)n - 1$
- $C_{mediu} = (1/4)n^2 + (1/4)n - 1/2$

Timp de rulare $T(n)$ - masurat in comparatii

- Exista un timp **minim** de rulare
- Exista un timp **maxim** de rulare
- $C_{min} = n - 1 \leq T(n) \leq (1/2)n^2 + (1/2)n - 1 = C_{max}$

Performanta

Comparatii

- $C_{min} = n - 1$
- $C_{max} = (1/2)n^2 + (1/2)n - 1$
- $C_{mediu} = (1/4)n^2 + (1/4)n - 1/2$

Timp de rulare $T(n)$ - masurat in comparatii

- Exista un timp **minim** de rulare
- Exista un timp **maxim** de rulare
- $C_{min} = n - 1 \leq T(n) \leq (1/2)n^2 + (1/2)n - 1 = C_{max}$

Timp de rulare $T(n)$ - masurat in alte operatii

- Comparatii + Mutari
- $C_{min} + M_{min} \leq T(n) \leq C_{max} + M_{max}$
- $3(n - 1) \leq T(n) \leq (1/2)n^2 + (3/2)n - 2$

Performanta

Comparatii

- $C_{min} = n - 1$
- $C_{max} = (1/2)n^2 + (1/2)n - 1$
- $C_{mediu} = (1/4)n^2 + (1/4)n - 1/2$

Timp de rulare $T(n)$ - masurat in comparatii

- Exista un timp **minim** de rulare
- Exista un timp **maxim** de rulare
- $C_{min} = n - 1 \leq T(n) \leq (1/2)n^2 + (1/2)n - 1 = C_{max}$

Timp de rulare $T(n)$ - masurat in alte operatii

- Comparatii + Mutari
- $C_{min} + M_{min} \leq T(n) \leq C_{max} + M_{max}$
- $3(n - 1) \leq T(n) \leq (1/2)n^2 + (3/2)n - 2$
- Fiecare linie de cod - in timp cst...

Margini superioare (si inferioare)... notatia asimptotica

Timp de rulare $T(n)$ - margine superioara

- $T(n) \leq (1/2)n^2 + (3/2)n - 2$
- $T(n) = \mathcal{O}(n^2)$

Margini superioare (si inferioare)... notatia asimptotica

Timp de rulare $T(n)$ - margine superioara

- $T(n) \leq (1/2)n^2 + (3/2)n - 2$
- $T(n) = \mathcal{O}(n^2)$

Timp de rulare $T(n)$ - margine inferioara

- $3(n-1) \leq T(n)$
- $T(n) = \Omega(n)$

Margini superioare (si inferioare)... notatia asimptotica

Timp de rulare $T(n)$ - margine superioara

- $T(n) \leq (1/2)n^2 + (3/2)n - 2$
- $T(n) = \mathcal{O}(n^2)$

Timp de rulare $T(n)$ - margine inferioara

- $3(n-1) \leq T(n)$
- $T(n) = \Omega(n)$

- abuz de notatie **!!!**

Margini superioare (si inferioare)... notatia asimptotica

Timp de rulare $T(n)$ - margine superioara

- $T(n) \leq (1/2)n^2 + (3/2)n - 2$
- $T(n) = \mathcal{O}(n^2)$

Timp de rulare $T(n)$ - margine inferioara

- $3(n-1) \leq T(n)$
- $T(n) = \Omega(n)$

- abuz de notatie **!!!**
- ... folositor

Margini superioare (si inferioare)... notatia asimptotica

Timp de rulare $T(n)$ - margine superioara

- $T(n) \leq (1/2)n^2 + (3/2)n - 2$
- $T(n) = \mathcal{O}(n^2)$

Timp de rulare $T(n)$ - margine inferioara

- $3(n-1) \leq T(n)$
- $T(n) = \Omega(n)$

- abuz de notatie **!!!**
- ... folositor

Timp de rulare $T(n)$ in cazul c. m. nefavorabil

- $T(n) = aC_{max} + b$ (exista cst $a, b > 0$)
- $T(n) = \Theta(n^2)$

Notatia asimptotica

- comportarea lui $T(n)$ cind $n \to \infty$

Notatia asimptotica

- comportarea lui $T(n)$ cind $n \to \infty$

Formal

- $f : \mathcal{N} \to \mathcal{R}_+$

Notatia asimptotica

- comportarea lui $T(n)$ cind $n \to \infty$

Formal

- $f : \mathcal{N} \to \mathcal{R}_+$ (f asimptotic pozitiva)

$$\mathcal{O}(g) := \{f \mid \exists c > 0, \exists n_0 \text{ a.i. } 0 \leq f(n) \leq cg(n) \text{ for all } n \geq n_0\}$$

Notatia asimptotica

- comportarea lui $T(n)$ cind $n \to \infty$

Formal

- $f : \mathcal{N} \to \mathcal{R}_+$ (f asimptotic pozitiva)

$$\mathcal{O}(g) := \{f \mid \exists c > 0, \exists n_0 \text{ a.i. } 0 \leq f(n) \leq cg(n) \text{ for all } n \geq n_0\}$$

$$\Omega(g) := \{f \mid \exists c > 0, \exists n_0 \text{ a.i. } 0 \leq cg(n) \leq f(n) \text{ for all } n \geq n_0\}$$

Notatia asimptotica

- comportarea lui $T(n)$ cind $n \to \infty$

Formal

- $f : \mathcal{N} \to \mathcal{R}_+$ (f asimptotic pozitiva)

$$\mathcal{O}(g) := \{f \mid \exists c > 0, \exists n_0 \text{ a.i. } 0 \leq f(n) \leq cg(n) \text{ for all } n \geq n_0\}$$

$$\Omega(g) := \{f \mid \exists c > 0, \exists n_0 \text{ a.i. } 0 \leq cg(n) \leq f(n) \text{ for all } n \geq n_0\}$$

$$\Theta(g) :=$$
$$\{f \mid \exists c_1, c_2 > 0, \exists n_0 \text{ a.i. } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \text{ for all } n \geq n_0\}$$

Notatia asimptotica

- comportarea lui $T(n)$ cind $n \to \infty$

Formal

- $f : \mathcal{N} \to \mathcal{R}_+$ (f asimptotic pozitiva)

$$\mathcal{O}(g) := \{f \mid \exists c > 0, \exists n_0 \text{ a.i. } 0 \le f(n) \le cg(n) \text{ for all } n \ge n_0\}$$

$$\Omega(g) := \{f \mid \exists c > 0, \exists n_0 \text{ a.i. } 0 \le cg(n) \le f(n) \text{ for all } n \ge n_0\}$$

$$\Theta(g) :=$$
$$\{f \mid \exists c_1, c_2 > 0, \exists n_0 \text{ a.i. } 0 \le c_1 g(n) \le f(n) \le c_2 g(n) \text{ for all } n \ge n_0\}$$

Properties:

$\Theta(g) = \mathcal{O}(g) \cap \Omega(g)$

Tranzitivitate

$f = \Theta(g), g = \Theta(h) \Longrightarrow f = \Theta(h)$
$f = \mathcal{O}(g), g = \mathcal{O}(h) \Longrightarrow f = \mathcal{O}(h)$
$f = \Omega(g), g = \Omega(h) \Longrightarrow f = \Omega(h)$

Reflexivitate

$f = \Theta(f)$
$f = \mathcal{O}(f)$
$f = \Omega(f)$

Simetrie

$f = \Theta(g)$ daca si numai daca $g = \Theta(f)$

Simetrie transpusa

$f = \mathcal{O}(g)$ daca si numai daca $g = \Omega(f)$

MergeSort

MergeSort

MergeSort(A, 1, n)

    if $n = 1$ sortat
    MergeSort(A, 1, n/2)
    MergeSort(A, n/2, n)
    Merge cei 2 subvectori ordonati

MergeSort

MergeSort

MergeSort(A, 1, n)
  if $n = 1$ sortat
  MergeSort(A, 1, n/2)
  MergeSort(A, n/2, n)
  Merge cei 2 subvectori ordonati

Merge = interclasare

- **input:** 2 vectori ordonati crescator $a[1..n]$ si $b[1..n]$
- **output:** vector $c[1..2n] = a \cup b$ ordonat crescator

MergeSort

MergeSort

MergeSort(A, 1, n)

    if $n = 1$ sortat
    MergeSort(A, 1, n/2)
    MergeSort(A, n/2, n)
    Merge cei 2 subvectori ordonati

Merge = interclasare

- **input:** 2 vectori ordonati crescator $a[1..n]$ si $b[1..n]$
- **output:** vector $c[1..2n] = a \cup b$ ordonat crescator
- $\Theta(n)$

MergeSort - performanta

MergeSort - ecuatia de recurenta pentru $T(n)$

- $T(n) = \Theta(1)$, daca $n = 1$
- $T(n) = 2T(n/2) + \Theta(n)$, daca $n > 1$

MergeSort - performanta

MergeSort - ecuatia de recurenta pentru $T(n)$

- $T(n) = \Theta(1)$, daca $n = 1$
- $T(n) = 2T(n/2) + \Theta(n)$, daca $n > 1$

- de rezolvat ecuatia ...

MergeSort - performanta

MergeSort - ecuatia de recurenta pentru $T(n)$

- $T(n) = \Theta(1)$, daca $n = 1$
- $T(n) = 2T(n/2) + \Theta(n)$, daca $n > 1$

- de rezolvat ecuatia ...
- arborele apelurilor recursive

MergeSort - performanta

MergeSort - ecuatia de recurenta pentru $T(n)$

- $T(n) = \Theta(1)$, daca $n = 1$
- $T(n) = 2T(n/2) + \Theta(n)$, daca $n > 1$

- de rezolvat ecuatia ...
- arborele apelurilor recursive

MergeSort - $T(n)$

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

MergeSort 'mai bun' decat InsertionSort

- MergeSort $T(n) = \Theta(nlogn)$
- InsertionSort $T(n) = \Theta(n^2)$
- $\Theta(nlogn)$ creste mai incet decat $\Theta(n^2)$

MergeSort 'mai bun' decat InsertionSort

- MergeSort $T(n) = \Theta(nlogn)$
- InsertionSort $T(n) = \Theta(n^2)$
- $\Theta(nlogn)$ creste mai incet decat $\Theta(n^2)$

- n 'mare' ?

MergeSort 'mai bun' decat InsertionSort

- MergeSort $T(n) = \Theta(nlogn)$
- InsertionSort $T(n) = \Theta(n^2)$
- $\Theta(nlogn)$ creste mai incet decat $\Theta(n^2)$

- n 'mare' ?
- pentru o aplicatie concreta, ce domeniu de valori are n?