

Curs 1: INTRODUCERE

Tehnici avansate de programare

Lect.dr. Iulia Banu
Departamentul de Informatică,
Universitatea din București

semestrul 1, 2017

Cuprins

- 1 Ore alocate
- 2 Conținut curs
- 3 Corectitudinea algoritmilor
- 4 Modalitate de evaluare

Întrebări, consultații sala **318**

iulia.banu@fmi.unibuc.ro

- curs 2h
- laborator 2h
- seminar 1h
- probleme/proiecte suplimentare, consultații marți 14-20

Algoritmi

corecți
eficienți

Algoritmi

corecți
eficienți

- identificarea structurilor de date și a tehnicilor potrivite;

Algoritmi

corecți
eficienți

- identificarea structurilor de date și a tehnicilor potrivite;
- analiza corectitudinii și a eficienței soluțiilor propuse;

Algoritmi

corecți
eficienți

- identificarea structurilor de date și a tehnicilor potrivite;
- analiza corectitudinii și a eficienței soluțiilor propuse;
- numeroase aplicații, exemple:

Algoritmi

corecți
eficienți

- identificarea structurilor de date și a tehnicilor potrivite;
- analiza corectitudinii și a eficienței soluțiilor propuse;
- numeroase aplicații, exemple:
 - Probleme de planificare optimă, folosire optimă a resurselor;

Algoritmi

corecți
eficienți

- identificarea structurilor de date și a tehnicilor potrivite;
- analiza corectitudinii și a eficienței soluțiilor propuse;
- numeroase aplicații, exemple:
 - Probleme de planificare optimă, folosire optimă a resurselor;
 - Programare jocuri (Branch and bound);

Algoritmi

corecți
eficienți

- identificarea structurilor de date și a tehnicilor potrivite;
- analiza corectitudinii și a eficienței soluțiilor propuse;
- numeroase aplicații, exemple:
 - Probleme de planificare optimă, folosire optimă a resurselor;
 - Programare jocuri (Branch and bound);
 - Geometrie computațională;

Algoritmi

corecți
eficienți

- identificarea structurilor de date și a tehnicilor potrivite;
- analiza corectitudinii și a eficienței soluțiilor propuse;
- numeroase aplicații, exemple:
 - Probleme de planificare optimă, folosire optimă a resurselor;
 - Programare jocuri (Branch and bound);
 - Geometrie computațională;
 - Cele mai apropiate două puncte dintr-o mulțime de puncte dată;

Algoritmi

corecți
eficienți

- identificarea structurilor de date și a tehnicilor potrivite;
- analiza corectitudinii și a eficienței soluțiilor propuse;
- numeroase aplicații, exemple:
 - Probleme de planificare optimă, folosire optimă a resurselor;
 - Programare jocuri (Branch and bound);
 - Geometrie computațională;
 - Cele mai apropiate două puncte dintr-o mulțime de puncte dată;
 - Procesare text;

Algoritmi

corecți
eficienți

- identificarea structurilor de date și a tehnicilor potrivite;
- analiza corectitudinii și a eficienței soluțiilor propuse;
- numeroase aplicații, exemple:
 - Probleme de planificare optimă, folosire optimă a resurselor;
 - Programare jocuri (Branch and bound);
 - Geometrie computațională;
 - Cele mai apropiate două puncte dintr-o mulțime de puncte dată;
 - Procesare text;
 - Căutare web, similitudini;

Algoritmi

corecți
eficienți

- identificarea structurilor de date și a tehnicilor potrivite;
- analiza corectitudinii și a eficienței soluțiilor propuse;
- numeroase aplicații, exemple:
 - Probleme de planificare optimă, folosire optimă a resurselor;
 - Programare jocuri (Branch and bound);
 - Geometrie computațională;
 - Cele mai apropiate două puncte dintr-o mulțime de puncte dată;
 - Procesare text;
 - Căutare web, similitudini;
- probleme/întrebări interviuri;

" People who think they know everything are a great annoyance to those of us who do"

"People who think they know everything are a great annoyance to those of us who do"

"The true delight is in the finding out rather than in the knowing."

Isaac Asimov

- Tehnici de programare:
 - Greedy
 - Divide et Impera
 - Programare Dinamica
 - Backtracking
 - Branch and Bound;
- Alte tipuri de algoritmi: algoritmi euristici, algoritmi probabiliști (Monte Carlo, Las Vegas);
- Algoritmi genetici
- Principiul lui Dirichlet;
- Analiza complexității unor algoritmi, NP-completitudine.

Demonstrarea corectitudinii

terminarea programului

Trebuie demonstrat că algoritmul se termină în timp finit.

Demonstrarea corectitudinii

terminarea programului

Trebuie demonstrat că algoritmul se termină în timp finit.

Fie E o expresie calculată în funcție de variabilele programului. Se poate alege un șir $\{x_n\}$ strict descrescător de numere naturale pozitive, unde $\{x_n\}$ este valoarea expresiei E la pasul n al algoritmului.

Demonstrarea corectitudinii

terminarea programului

Trebuie demonstrat că algoritmul se termină în timp finit.

Fie E o expresie calculată în funcție de variabilele programului. Se poate alege un șir $\{x_n\}$ strict descrescător de numere naturale pozitive, unde $\{x_n\}$ este valoarea expresiei E la pasul n al algoritmului.

corectitudine parțială

Presupunând că algoritmul se termină, rezultatul obținut la final este cel corect.

Demonstrarea corectitudinii

terminarea programului

Trebuie demonstrat că algoritmul se termină în timp finit.

Fie E o expresie calculată în funcție de variabilele programului. Se poate alege un șir $\{x_n\}$ strict descrescător de numere naturale pozitive, unde $\{x_n\}$ este valoarea expresiei E la pasul n al algoritmului.

corectitudine parțială

Presupunând că algoritmul se termină, rezultatul obținut la final este cel corect.

Demonstrarea corectitudinii

terminarea programului

Trebuie demonstrat că algoritmul se termină în timp finit.

Fie E o expresie calculată în funcție de variabilele programului. Se poate alege un șir $\{x_n\}$ strict descrescător de numere naturale pozitive, unde $\{x_n\}$ este valoarea expresiei E la pasul n al algoritmului.

corectitudine parțială

Presupunând că algoritmul se termină, rezultatul obținut la final este cel corect.

Invarianti = relații ce trebuie îndeplinite la orice trecere a programului prin acel loc.

Exemplu

Metoda de înmulțire a țăranului rus:

```
x ← a; y ← b; p ← 0;  
while x > 0  
    { xy + p = ab } (*)  
    if x impar then p ← p + y;  
    x ← x div 2;  
    y ← y + y;  
write(p);
```

Costul spațiului

Timpul de execuție

$T(n)$ = timpul de executare pentru orice set de date de intrare de dimensiune n .

Definiție (Notăție asimptotică)

Fie $f : \mathbb{N} \rightarrow \mathbb{R}_+$

$$\mathcal{O}(g) := \{f | \exists c > 0, \exists n_0 \in \mathbb{N} \text{ a.i } 0 \leq f(n) \leq cg(n) \forall n > n_0\}$$

$$\Omega(g) := \{f | \exists c > 0, \exists n_0 \in \mathbb{N} \text{ a.i } 0 \leq cg(n) \leq f(n) \forall n > n_0\}$$

$$\Theta(g) := \{f | \exists c_1, c_2 > 0, \exists n_0 \in \mathbb{N} \text{ a.i}$$

$$0 \leq c_1g(n) \leq f(n) \leq c_2g(n) \forall n > n_0\}$$

Exemplu: Să se determine minimul și maximul elementelor unui vector.

Exemplu: Să se determine minimul și maximul elementelor unui vector.

Orice algoritm corect va efectua cel puțin $\lceil 3n/2 \rceil - 2$ comparații.

Exemplu: Să se determine minimul și maximul elementelor unui vector.

Orice algoritm corect va efectua cel puțin $\lceil 3n/2 \rceil - 2$ comparații.

```
if n impar then m  $\leftarrow$   $a_1$ ; M  $\leftarrow$   $a_1$ ; k  $\leftarrow$  2;  
else if  $a_1 < a_2$  then m  $\leftarrow$   $a_1$ ; M  $\leftarrow$   $a_2$ ;  
      else m  $\leftarrow$   $a_2$ ; M  $\leftarrow$   $a_1$ ;  
      k  $\leftarrow$  3;
```

Exemplu: Să se determine minimul și maximul elementelor unui vector.

Orice algoritm corect va efectua cel puțin $\lceil 3n/2 \rceil - 2$ comparații.

```
if n impar then m  $\leftarrow$   $a_1$ ; M  $\leftarrow$   $a_1$ ; k  $\leftarrow$  2;
else if  $a_1 < a_2$  then m  $\leftarrow$   $a_1$ ; M  $\leftarrow$   $a_2$ ;
      else m  $\leftarrow$   $a_2$ ; M  $\leftarrow$   $a_1$ ;
      k  $\leftarrow$  3;
while k  $\leq$  n-1
  if  $a_k < a_{k+1}$  then
    if  $a_k < m$  then m  $\leftarrow$   $a_k$ ;
    if  $a_{k+1} > M$  then M  $\leftarrow$   $a_{k+1}$ ;
  else
    if  $a_{k+1} < m$  then m  $\leftarrow$   $a_{k+1}$ ;
    if  $a_k > M$  then M  $\leftarrow$   $a_k$ ;
  k  $\leftarrow$  k+2;
```

Săptămâna 14

- test laborator **50%**
- activitate **50%**
 - verificare curs **60%** exemple întrebări moodle + **10%** oficiu
 - 3 teme laborator **30%**

! Nota la testul de laborator minim 5.

- seminar 1p bonus adaugat la nota finala
- puncte bonus pentru activitate la curs adaugate la nota finala

- Horia Georgescu. Tehnici de programare. Editura Universității din București 2005
- Leon Livovschi, Horia Georgescu. Sinteza și analiza algoritmilor. 1986
- T.H. Cormen, C.E. Leiserson, R.R. Rivest Introducere în algoritmi, Mit Press, trad. Computer Libris Agora
- Jon Kleinberg, Éva Tardos, Algorithm Design, Addison-Wesley 2005
<http://www.cs.princeton.edu/~wayne/kleinberg-tardos/>
- S. Dasgupta, C.H. Papadimitriou, U.V. Vazirani, Algorithms, McGraw-Hill, 2008
- <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/>
- <http://moodle.fmi.unibuc.ro/course/>