

Probleme Curs 1, Curs 2 SDA

Probleme propuse pentru analiza complexitatii, liste implementate inlantuit, sau secvential, stiva, coada. Unele din problemele propuse sunt preluate de pe [Geeks-ForGeeks](#).

Probleme Complexitate

Problema 1 Pentru algoritmi din secventele de mai jos, calculati numarul de pasi efectuati de buclele (sau apelurile recursive) din fiecare algoritm (in functie de N, M). Care este complexitatea in timp a algoritmilor?

Algorithm 1 First Algorithm

```
1: procedure P1( $N, M$ )
2:    $a, b \leftarrow 0$ 
3:   for  $i \leftarrow 1$  to  $N$  do
4:      $a \leftarrow a + 10$ 
5:   for  $j \leftarrow 1$  to  $M - 1$  do
6:      $b \leftarrow b + 20$ 
```

Algorithm 2 Second Algorithm

```
1: procedure P2( $N, M$ )
2:    $a, b \leftarrow 0$ 
3:    $c \leftarrow 1$ 
4:    $e \leftarrow 0.1$ 
5:   for  $i \leftarrow 1$  to  $N - 1$  do
6:      $a \leftarrow a + 10$ 
7:      $c \leftarrow 2 * c$ 
8:   for  $j \leftarrow 1$  to  $M - 1$  do
9:      $b \leftarrow b + 20$ 
10:     $e \leftarrow e * 10.5$ 
```

Author(s): Raluca Brehar, Camelia Lemnaru

Algorithm 3 Third Algorithm

```
1: procedure P3(N)
2:    $a \leftarrow 0$ 
3:   for  $i \leftarrow 1$  to  $N$  do
4:     for  $j \leftarrow N$  downto  $i$  do
5:        $a \leftarrow a + i + j$ 
```

Algorithm 4 Fourth Algorithm

```
1: procedure P4(N)
2:    $a \leftarrow 0$ 
3:   for  $i \leftarrow N/2$  to  $N$  do
4:      $j \leftarrow 1$ 
5:     while  $j \leq N$  do
6:        $a \leftarrow a + N/2$ 
7:        $j \leftarrow j * 2$ 
```

Algorithm 5 Fifth Algorithm

```
1: procedure P5(N)
2:    $a \leftarrow 0$ 
3:   for  $i \leftarrow N/2$  to  $N$  do
4:      $j \leftarrow 1$ 
5:     while  $j \leq N$  do
6:        $a \leftarrow a + N/2$ 
7:        $j \leftarrow j + 2$ 
```

Algorithm 6 Sixth Algorithm

```
1: procedure P6(N)
2:    $a \leftarrow 0$ 
3:    $i \leftarrow N$ 
4:   while  $i > 0$  do
5:      $a \leftarrow a + i$ 
6:      $i \leftarrow i/2$ 
```

Algorithm 7 Seventh Algorithm

```
1: procedure P7(N)
2:    $a \leftarrow 0$ 
3:    $i \leftarrow N$ 
4:   while  $i > 0$  do
5:      $a \leftarrow a + i$ 
6:      $i \leftarrow i - 2$ 
```

Algorithm 8 Eighth Algorithm

```

1: procedure P8(N)
2:   if  $N \leq 1$  then return 1
3:    $x \leftarrow 0$ 
4:   for  $i \leftarrow 1$  to  $N$  do
5:      $x \leftarrow x + P8(N - 1)$ 

```

Algorithm 9 Ninth Algorithm

```

1: procedure P9(N)
2:   if  $N \leq 1$  then return 1
   return  $P9(N - 1) + P9(N - 1)$ 

```

Probleme Liste

Problema 2 Se da o lista simplu inlantuita, cu referinta la inceputul listei. Descrieti, in pseudocod, un algoritm care numara elementele duplicat din lista. (Un element este duplicat daca apare de cel putin 2 ori in lista). Analizati complexitatea algoritmului propus.

Explicatie. Lista: $3 \rightarrow 7 \rightarrow 7 \rightarrow 2 \rightarrow 3 \rightarrow 26 \rightarrow 7 \rightarrow 1 \rightarrow \text{NULL}$;

Lista are doua elemente duplicat: 3 si 7;

Problema 3 Se da o lista simplu inlantuita, cu referinta la inceputul listei. Descrieti, in pseudocod, un algoritm care inlocuieste toate valorile unicat din lista, cu o valoare data (repl). Analizati complexitatea algoritmului propus.

Explicatie. Lista: $3 \rightarrow 7 \rightarrow 7 \rightarrow 2 \rightarrow 3 \rightarrow 26 \rightarrow 7 \rightarrow 1 \rightarrow \text{NULL}$;

Valoarea de inlocuire: 15

Lista rezultata:

$3 \rightarrow 7 \rightarrow 7 \rightarrow 15 \rightarrow 3 \rightarrow 15 \rightarrow 7 \rightarrow 15 \rightarrow \text{NULL}$;

Problema 4 Se da o lista simplu inlantuita, cu referinta la inceputul listei. Descrieti, in pseudocod, un algoritm care inverseaza lista. Analizati complexitatea algoritmului propus.

Explicatie. Lista: $3 \rightarrow 7 \rightarrow 17 \rightarrow 2 \rightarrow 3 \rightarrow 26 \rightarrow \text{NULL}$;

Lista rezultata:

Lista: $26 \rightarrow 3 \rightarrow 2 \rightarrow 17 \rightarrow 7 \rightarrow 3 \rightarrow \text{NULL}$;

Problema 5 Se da o lista simplu inlantuita, cu referinta la inceputul listei. Descrieti, in pseudocod, un algoritm care sterge ultima aparitie a unui element. Analizati complexitatea algoritmului propus.

Explicatie. Lista: $3 \rightarrow 7 \rightarrow 3 \rightarrow 17 \rightarrow 2 \rightarrow 3 \rightarrow 26 \rightarrow \text{NULL}$;

Elementul de sters: 3

Lista rezultat: $3 \rightarrow 7 \rightarrow 3 \rightarrow 17 \rightarrow 2 \rightarrow 26 \rightarrow \text{NULL}$;

Problema 6 Se da o lista simplu inlantuita ORDONATA, cu referinta la inceputul listei. Descrieti, in pseudocod, un algoritm care insereaza un element dat in lista (lista rezultat trebuie sa fie si ea ordonata). Analizati complexitatea algoritmului propus.

Explicatie. Lista:

$3 \rightarrow 7 \rightarrow 13 \rightarrow 17 \rightarrow 25 \rightarrow 31 \rightarrow 126 \rightarrow \text{NULL}$;

Element de inserat: 28

Lista rezultat:

$3 \rightarrow 7 \rightarrow 13 \rightarrow 17 \rightarrow 25 \rightarrow 28 \rightarrow 31 \rightarrow 126 \rightarrow \text{NULL}$;

Problema 7 Se da o lista simplu inlantuita, cu referinta la inceputul listei. Lista contine elemente numere intregi. Descrieti, in pseudocod, un algoritm care creaza o noua lista care are ca si campuri elementele si frecventa lor de aparitie. Analizati complexitatea algoritmului propus.

Explicatie. E.g. data lista initiala contine elementele:

$1 \rightarrow 3 \rightarrow 3 \rightarrow 2 \rightarrow 15 \rightarrow 6 \rightarrow 15 \rightarrow 6 \rightarrow 15 \rightarrow 10 \rightarrow \text{NULL}$;

Lista cu frecventele:

$1:1 \rightarrow 3:2 \rightarrow 2:1 \rightarrow 15:3 \rightarrow 6:2 \rightarrow 10:1 \rightarrow \text{NULL}$;

Problema 8 Se da o lista simplu inlantuita, cu referinta la inceputul listei. Descrieti, in pseudocod, un algoritm care roteste lista (circular) cu K pozitii spre stanga. Analizati complexitatea algoritmului propus.

Explicatie. E.g. data lista initiala contine elementele $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$, si $K = 4$, lista rezultat arata asa: $5 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$.

Problema 9 Se da o lista simplu inlantuita, cu referinta la inceputul listei. Lista contine elemente numere naturale ≤ 9 . Descrieti, in pseudocod, un algoritm care determina daca lista este palindromica. Analizati complexitatea algoritmului propus.

Explicatie. E.g. Lista $5 \rightarrow \text{NULL}$; Este palindromica.

Lista: $4 \rightarrow 9 \rightarrow \text{NULL}$; Nu este palindromica.

Lista: $2 \rightarrow 7 \rightarrow 7 \rightarrow 2 \rightarrow \text{NULL}$; Este palindromica.

Lista: $3 \rightarrow 7 \rightarrow 7 \rightarrow 2 \rightarrow \text{NULL}$; Nu este palindromica.

Problema 10 Se da o lista simplu inlantuita, cu referinta la inceputul listei. Lista contine elemente intregi. Descrieti, in pseudocod, un algoritm care printr-o singura parcurgere a listei determina elementul din mijlocul listei. Analizati complexitatea algoritmului propus.

Explicatie. E.g. Lista $5 \rightarrow \text{NULL}$; Elementul din mijloc este: 5

Lista: $45 \rightarrow 39 \rightarrow \text{NULL}$ Elementul din mijloc este: 39

$32 \rightarrow 41 \rightarrow 25 \rightarrow \text{NULL}$ Elementul din mijloc este: 41

$27 \rightarrow 32 \rightarrow 49 \rightarrow 51 \rightarrow \text{NULL}$ Elementul din mijloc este: 49

Problema 11 Se dau doua liste simplu inlantuite ordonate crescator, fiecare avand referinta la inceputul listei. Listele contin elemente intregi. Descrieti, in pseudocod, un algoritm care interclaseaza cele doua liste. Analizati complexitatea algoritmului propus.

Explicatie. Lista (a): $32 \rightarrow 41 \rightarrow 52 \rightarrow \text{NULL}$

Lista (b): $27 \rightarrow 39 \rightarrow 49 \rightarrow 151 \rightarrow \text{NULL}$

Lista care rezulta dupa interclasare:

$27 \rightarrow 32 \rightarrow 39 \rightarrow 41 \rightarrow 49 \rightarrow 52 \rightarrow 151 \rightarrow \text{NULL}$

Problema 12 Se dau doua liste simplu inlantuite, fiecare avand referinta la inceputul listei. Listele contin elemente intregi. Descrieti, in pseudocod, un algoritm care determina lista care reprezinta intersectia celor doua liste. Analizati complexitatea algoritmului propus.

Explicatie. Lista (a): $32 \rightarrow 41 \rightarrow 52 \rightarrow \text{NULL}$

Lista (b): $27 \rightarrow 41 \rightarrow 49 \rightarrow 52 \rightarrow \text{NULL}$

Lista care reprezinta intersectia:

$41 \rightarrow 52 \rightarrow \text{NULL}$

Probleme Stiva, Coadă

Problema 13 Descrieti un algoritm pentru inversarea unei cozi, utilizand recursivitatea. Nu este permisa utilizarea buclor (while, for..etc), doar a operatiilor fundamentale pe coada (enqueue(), dequeue(), size(), front()). Analizati complexitatea algoritmului propus.

Problema 14 Descrieti un algoritm pentru inversarea unei stive, utilizand recursivitatea. Nu este permisa utilizarea buclor (while, for..etc), doar a operatiilor fundamentale pe stiva (push(), pop(), size(), top()). Analizati complexitatea algoritmului propus.

Problema 15 Descrieti un algoritm pentru stergerea elementului din mijlocul unei stive, utilizand recursivitatea. Nu este permisa utilizarea buclor (while, for..etc), doar a operatiilor fundamentale pe stiva (push(), pop(), size(), top()). Analizati complexitatea algoritmului propus.

Explicatie. Stiva initiala contine:

(top) 32, 41, 25, 27, 49, 152, 102

Stiva dupa stergerea mijlocului:

(top) 32, 41, 25, 49, 152, 102

Problema 16 Descrieti un algoritm pentru interclasarea elementelor din prima jumătate a unei cozi cu elementele din a doua jumătate. Coadă are număr par de elemente. Utilizati o stiva ca structura auxiliara. Analizati complexitatea algoritmului propus.

Explicatie. Coadă:

32, 41, 25, 27, 141, 49, 152, 102

Coadă după interclasarea celor două jumătăți:
32, 141, 41, 49, 25, 152, 27, 102