



Structuri de date

Rotaru Alexandru Andrei
rotarualexandruandrei94@gmail.com
2019-2020

University Politehnica of Bucharest

1 Precizari

Laborantul va preciza ce exercitii veti avea de rezolvat. Pentru rezolvarea fiecarui exercitiu veti folosi un fisier separat. **Optional**, va puteti verifica folosind checker-ul (daca este pus la dispozitie).

2 Resurse

2.1 Vizualizare

Pentru a putea vizualiza o parte din algoritmi si intelege mai bine cum functioneaza structurile de date accesati

<https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>

2.2 Link-uri catre platforme online cu probleme

hackerrank: <https://www.hackerrank.com/domains/data-structures>

leetcode: <https://leetcode.com/problemset/all/>

2.3 Tool-uri de debug

- gdb: <https://cs.baylor.edu/~donahoo/tools/gdb/tutorial.html>
- valgrind: <http://valgrind.org/docs/manual/quick-start.html>

2.4 Feedback

Pentru a semnala probleme sau a oferi sugestii pentru laboratoarele urmatoare:

<https://forms.gle/8bRw7mPgqaJ9tRzd6>

3 Exercitii

3.1 Exercițiul 1 [2 puncte]

Sa se scrie o functie recursiva pentru numararea valorilor dintr-un vector care sunt egale cu o valoarea data.

Listing 1: Semnatura functiei

```
1 int count_equals(int value, int *vector,
2   size_t vector_length);
```

Input vector = [2, 5, 7, 9, 5, 6, 5, 5, 11]; value = 5;

Output 4

3.2 Exercițiul 2 [2 puncte]

Sa se implementeze o functie recursiva care calculeaza al k-lea terment fibonacci.

Listing 2: Semnatura functiei

```
1 int fibonacci(int k);
```

Input 10

Output 55

3.3 Exercițiul 3 [2 puncte]

Sa se implementeze o functie recursiva care calculeaza ridicarea la putere a unui numar natural. Pentru a realiza acest lucru folositi urmatoarele formule:

$$base^{power} = base^{power/2} * base^{power/2}, \text{daca } power \text{ par} \quad (1)$$

$$base^{power} = base * base^{power/2} * base^{power/2}, \text{daca } power \text{ impar} \quad (2)$$

Listing 3: Semnatura functiei

```
1 double power(int base, int pow);
```

Input base=2 exponential=3

Output 8

3.4 Exercițiul 4 [2 puncte]

Sa se implementeze o functie ce genereaza un sir de numere pseudo-aleatoare in intervalul $[0, 1]$. Numerele aleatoare se vor genera folosind formula:

$$x_{n+1} = r\dot{x}_n(1 - x_n), x_0 = 0.45, r = 3.75 \quad (3)$$

Listing 4: Semnatura functiei

```
1 double nth_random_value(size_t n);
```

Primele 6 elemente incluzand valoarea initiala: $[0.45, 0.92, 0.25, 0.70, 0.78, 0.63]$

3.5 Exercițiul 5 [3 puncte]

Sa se implementeze o functie care primeste ca parametru un vector de n elemente naturale pozitive. Asupra acestui vector se vor aplica urmatoarea operatie: vectorul se imparte in 2 si daca are lungime impara se elimina elementul din mijloc apoi se apeleaza aceasi procedura pentru fiecare din cele 2 jumatati. Procedura se opreste cand vectorul are lungimea 1.

Listing 5: Semnatura functiei

```
1 void process(int *vector, size_t *vector_len);
```

Input vector=[1, 2, 3, 4, 5, 6, 7]

Output 1, 3, 5, 7