

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S2-PP-CA-CB-CC](#) / [13 March - 19 March](#) / [Test - Laborator 2](#)

Started on Tuesday, 21 March 2023, 2:09 PM

State Finished

Completed on Tuesday, 21 March 2023, 2:15 PM

Time taken 5 mins 58 secs

Marks 3.00/4.00

Grade 7.50 out of 10.00 (75%)

Question 1

Correct

Mark 1.00 out of 1.00

De ce recursivitatea arborescentă este inefficientă?

Select one:

- ☒ a. Din cauza timpului consumat de numărul mare de apeluri recursive lansate și din cauza spațiului consumat de informația depusă pe stivă, necesară pentru revenirea din recursivitate ✓
- ☐ b. Pentru că se folosește memoizarea apelurilor recursive
- ☐ c. Deoarece are adesea complexitate spațială exponențială
- ☐ d. Deoarece avem apeluri recursive unice, fapt care duce la stocarea multor apeluri pe stivă

Răspunsul dumneavoastră este corect.

Din cauza timpului consumat de numărul mare de apeluri recursive lansate și din cauza spațiului consumat de informația depusă pe stivă, necesară pentru revenirea din recursivitate

The correct answer is:

Question 2

Correct

Mark 1.00 out of 1.00

Ce complexitate spațială (spațiu suplimentar pe stivă, nu cel ocupat de lista de intrare) are funcția `inc` definită mai jos?

```
(define (inc L)
  (if (null? L)
      '()
      (cons (+ (car L) 1) (inc (cdr L)))))
```

Notăm cu n numărul de elemente dintr-o listă oarecare L .

Select one:

- ☒ a. $\Theta(n)$ ✓
- ☐ b. $\Theta(1)$
- ☐ c. $\Theta(n^2)$
- ☐ d. $\Theta(n^3)$

Răspunsul dumneavoastră este corect.

The correct answer is: $\Theta(n)$

Question 3

Incorrect

Mark 0.00 out of 1.00

Ce tip de recursivitate produce implementarea algoritmului mergesort (sortăm recursiv prima și a doua jumătate a vectorului, apoi interclasăm rezultatele)?

Select one:

- ☐ a. Recursivitate arborescentă
- ☐ b. Recursivitate pe stivă
- ☐ c. Recursivitate pe coadă
- ☒ d. Putem implementa cu orice tip de recursivitate ✗

Răspunsul dumneavoastră este incorect.

The correct answer is: Recursivitate arborescentă

Question 4

Correct

Mark 1.00 out of 1.00

Dacă dorim să implementăm o funcție recursivă, știind că va fi folosită pe date de intrare de dimensiune foarte mare, atunci ce tip de recursivitate ar trebui folosit pentru a nu avea probleme de tip stack overflow?

Select one:

- ☒ a. Recursivitate pe coadă ✓
- ☐ b. Recursivitate pe stivă
- ☐ c. Recursivitate arborescentă
- ☐ d. Orice tip de recursivitate, cât timp găsim un algoritm de complexitate temporală polinomială

Răspunsul dumneavoastră este corect.

The correct answer is: Recursivitate pe coadă

[◀ Test - Laborator 1](#)[Test - Laborator 3 ▶](#)