

# ИСПИТ АИПС ТЕОРИЈА

База на прашања :d

Link do drugata baza:

[APS teorija - Google Docs](#)

Операција	Подредениа поврзана листа	Неподредена поврзана листа	Подредена низа	Неподредена низа
додај	$O(n)$	$O(1)$	$O(n)$	$O(1)$
извади	$O(1)$	$O(n)$	$O(1)$	$O(n)$

[APS teorija](#)

- Која е сложеноста на алгоритмот?

Question 1  
Not yet answered  
Marked out of 2.00  
[Flag question](#)

Da se напише функцијата за бројот на извршени операции за дадениот псевдо код и да се пресмета асимптотската горна граница "големо О" за истиот код:

```
simple_computer(x, n)
{
    int sum=0;
    for (int i=0; i<n; i++)
        sum+=2*i*x[i];
    return sum;
}
```

Select one:

a.  $f(n)=2n^2n+4, O(n^2n)$

b.  $f(n)=5n+8, O(n)$

c.  $f(n)=3n+6, O(3n)$

d.  $f(n)=2n+8, O(n)$

-O(N)

Да се напише функцијата за бројот на извршени операции за дадениот псевдо код и .."Големо О" за истиот код:

- **5n + 8**

Колкава е сложеноста на алгоритмот со кој се прави спојување на две сортирани листи со големина  $m$  и  $n$  во сортирана листа со големина  $m + n$ .

Select one:

- a.  $O(\log m + \log n)$
- b.  $O(n)$
- c.  $O(m)$
- d.  $O(m+n)$

- **$O(m+n)$**

Question 2

Not yet  
answered

Marked out of  
1.00

Flag question

Што се случува кога top-down пристап на динамичко програмирање ќе се примени на било кој проблем?

Select one:

- a. Се зголемува мемориската комплексност, но се намалува временската комплексност
- b. Се зголемува временската комплексност, но се намалува мемориската комплексност
- c. Се намалуваат и мемориската и временската комплексност
- d. Се зголемуваат и мемориската и временската комплексност

- Се зголемува мемориската комплексност, но се намалува временската комплексност

Што се случува кога top-down пристап на динамичко програмирање ќе се примени на било кој проблем?

- **A. Се зголемува мемориската комплексност, но се намалува временската комплексност**

ITMI И ПОДАТОЧНИ СТРУКТУРИ-2020/2021/Z

/courses / Апли-2020/2021/Z-36\_30636 / Испит Јуни 10.6.2021 / Испит Јуни - термин 1

Question 3  
Not yet answered  
Marked out of 1.00  
Flag question

Kој алгоритам за сортирање ќе се изврши за најкратко време во случај кога сите елементи од влезната низа се идентични? Претпоставете ги типичните имплементации на алгоритмите за сортирање.

Select one:

- INSERTION SORT
- HEAP SORT
- MERGE SORT
- SELECTION SORT

Next page

Кој алгоритам за сортирање ќе се изврши за најкратко време во случај кога сите елементи од влезната низа се идентични? .. сортирање

- ### - Insertion sort

Кој е излезот после дадена листа на операции ( ако при pop операторот се печати елементот кој е ?повратна? вредност )

85251

СТРУКТУРИ / Испит - јануари / Испит јануари - термини

Со која техника на програмирање може ефикасно да се најде најдолга растечка подниза на дадена низа?

Select one:

- a. Техники со враќање наназад
- b. Лакоми алгоритми
- c. Динамичко програмирање
- d. Груба сила

Question 3

incorrect

Mark 0.00 out  
of 1.00

Flag  
question

Edit  
question

Со која техника на програмирање може ефикасно да се најде најдолга растечка подниза на дадена низа?

Select one:

- a. Лакоми алгоритми
- b. Динамичко програмирање
- c. Техники со враќање наназад X
- d. Груба сила

Your answer is incorrect.

The correct answer is: Динамичко програмирање

Со која техника на програмирање може ефикасно да се најде најдолга растечка подниза на дадена низа?

- **Динамичко програмирање**

2222.

The screenshot shows a question from an online exam. At the top, it says "Динамичкото програмирање како архетип се заснова на" (Dynamic programming as a paradigm is based on) and "Select one:". Below this is a list of four options:

- a. паметење на пресметани резултати и генерирање на простор на резултати
- b. Преполување на влезно множество на податоци
- c. комбинација на алчни алгоритми и груба сила
- d. рекурзија и брзи математички пресметки

Динамичкото програмирање како архетип се заснова на:

- **А. паметење на пресметани резултати и генерирање на простор на резултати.**

Што од наведеното е точно за стек? 

Select one:

- a. Имплицитно се имплементира со рекузија
- b. Имплицитно се имплементира со итерација
- c. Може да се имплементира само со двојно поврзана листа
- d. Користи приоритет

Што од наведеното е точно за стек?

- a. Имплицитно се имплементира со рекурзија

Нека е дадена следната низа од елементи:

20, 47, 15, 8, 9, 4, 40, 30, 12, 17-

Ако се примени merge sort алгоритмот за сортирање на оваа низа, како ќе изгледа низата после второто изминување (second pass) од алгоритмот?

- 8, 15, 20, 47, 4, 9, 30, 40, 12, 17

Question 7

Not yet

answered

Marked out of

0

Flag question

Доврши ја реченицата да биде точна: Вметнување елемент е потешко ...

Select one:

a.

ниту едно од понуденото

b.

...на почеток на еднострano поврзана листа, отколку на почеток на низа ( $A[0]$ )

c.

.. на крај на низа ( $A[n]$ ,  $n$  е познато) отколку на крај на еднострano поврзана листа

d. ...на почеток на низа ( $A[0]$ ) отколку на почеток на еднострano поврзана листа

Доврши ја реченицата да биде точна:

Вметнување елемент е потешко ....

D. ... на почеток на низа ( $A[0]$ ) отколку на почеток на еднострano поврзана листа

Question 6

Not yet

answered

Marked out of  
1.00

Flag question

Колку стека се потребни за да се имплементира редица

Select one:

a. 2

b. 4

c. 1

d. 3

Колку стека се потребни за да се имплементира редица

- 2

Question 3  
Not yet  
answered  
Marked out of  
1.00  
Flag question

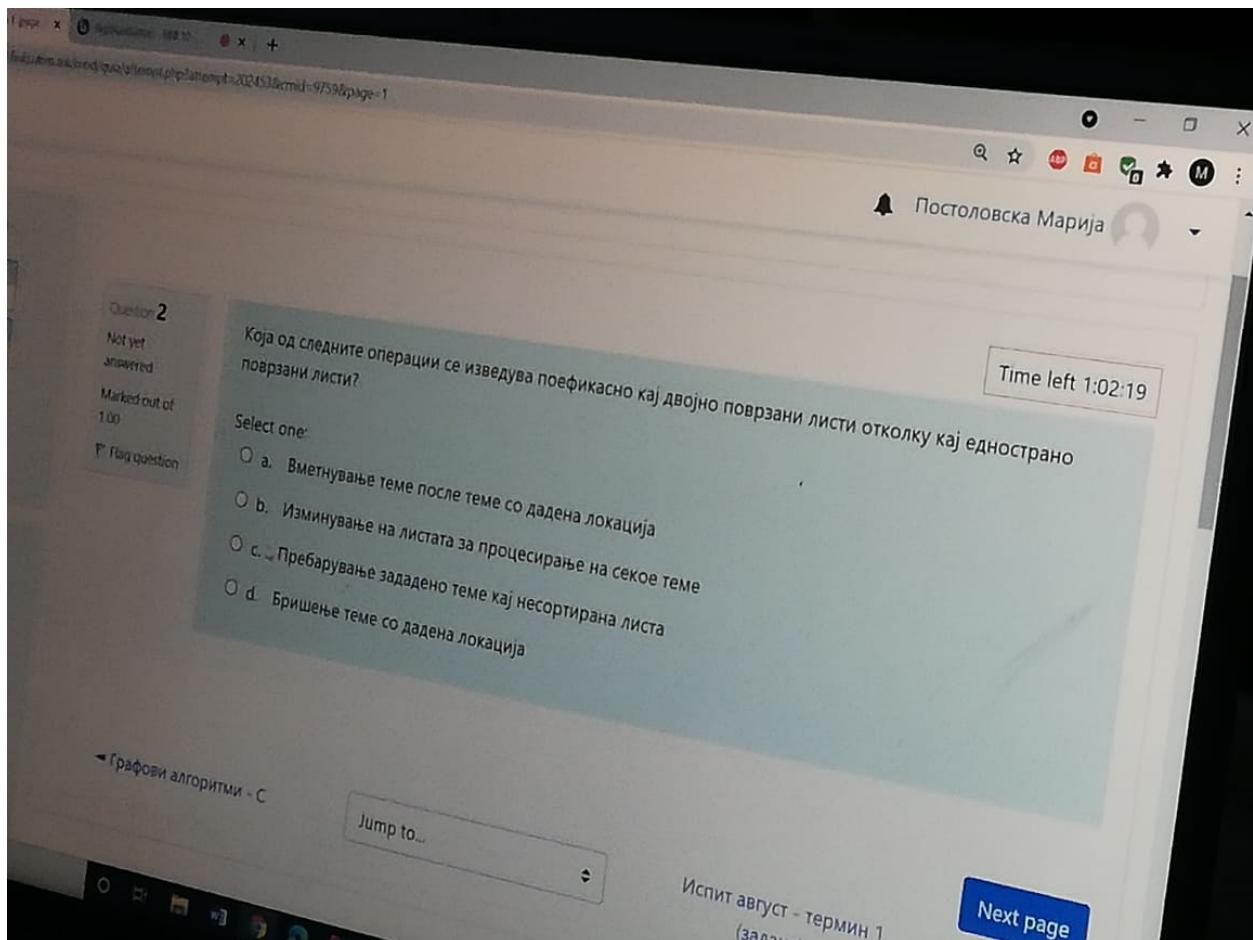
Кој алгоритам за сортирање ќе се изврши за најкратко време во случај кога сите елементи од влезната низа се идентични? Претпоставете ги типичните имплементации на алгоритмите за сортирање.

Select one:

- HEAP SORT
- SELECTION SORT
- INSERTION SORT
- MERGE SORT

Кој алгоритам на сортирање ќе се изврши за најкратко време во случај кога сите елементи од влезната низа се идентични? Претпоставете ги типичните имплементации на алгоритмите за сортирање.

- **Insertion Sort**



Која од следните операции се изведува поефикасно кај двојно поврзани листи отколку кај еднострено поврзани листи?

Select one:

- a. Вметнување теме после теме со дадена локација
- b. Изминување на листата за процесирање на секое теме
- c. Пребарување зададено теме кај несортирана листа
- d. Бришење теме со дадена локација

При имплементација на стек со поврзана листа, додавањето и вадењето на елемент се врши

- На почетокот на листата бидејќи бришење + додавање ќе бидат о(1)

The screenshot shows a Moodle quiz interface. At the top, it says "Изпити и структури-2020/2021/Z" and "Испит Септември 27.8.2021 Термин 2 09:30-11:00 / Испит август - термин 2". A "Time left" timer is visible in the top right corner. The question title is "Архетипот на алчни алгоритми". Below it, the instruction "Select one" is followed by four options:

- a. наоѓа точно решение за одреден тип на проблем
- b. наоѓа најбрзо решение за одреден тип на проблем
- c. секогаш наоѓа точно решение за било кој тип на проблем
- d. наоѓа најбрзо решение за било кој тип на проблем

Below the options are "Clear my choice" and "Jump to...". In the bottom right corner, there is a link "Испит август - термин 2 (задаци)".

#### Архетипот на алчни алгоритми

- **B, (B => In computer science, a greedy algorithm is an algorithm that finds a solution to problems in the shortest time possible )**

Замислете дека постојат две решенија на еден проблем. Првото решение има последователни рекурзивни повика кои го преполовуваат бројот на почтени елементи.

Select one:

a. 3  
 b. 1  
 c. 2  
 d. Нема степен

Clear my choice

Question 3  
Not yet answered  
Marked out of 1.00  
Flag question

Задислете дека постојат две решенија на еден проблем. Првото решение има два последователни рекурзивни повика кои го преполовуваат бројот на почетни елементи. Второто решение ги изменува половина од елементите во циклус. Кое од решенијата е поефикасно и зошто?

Select one:

a. Второто затоа што го решава проблемот само со половина од елементите, додека првото непотребно два пати ги третира елементите.  
 b. Првото, затоа што има логоритамска комплексност за разлика од второто кое има линеарна комплексност.  
 c. Првото затоа што има логоритамска комплексност за разлика од второто што има половично линеарна комплексност.  
 d. Првото затоа што рекурзија е секогаш поефикасна од циклус.

Clear my choice

Finish attempt →

Второто решение ги изменува половина од елементите во циклус. Кое од решенијата е поефикасно и зошто?

- **b**

# Структури-2020/2021/Z

/ Испит Септември 27.8.2021 Термин 2 09:30-11:00 / Испит август - термин 2

Time left 0:58:

За дадена низа  $a[left..right]$ , во кој алгоритам се спречува следната инструкција (во псевдојазик)

копирај  $a[x+1..right]$  во  $a[x..right-1]$

Select one:

- a. Ниту еден од поизведените одговори
- b. Копирање една низа во друга
- c. Бришење елемент  $a[x]$
- d. Додавање елемент  $a[x]$

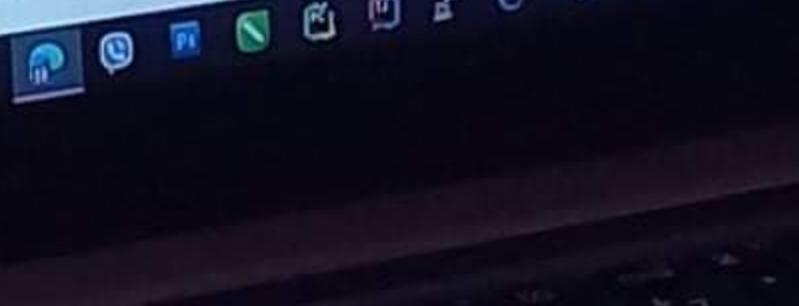
No

Испит август - термин 2

(започнато)

Jump to:

13.



За дадена низа  $a[]$  во кој алгоритам се спречнува следната инструкција ( во псевдо java)  
- бришење+

**Question 6**Not yet  
answeredMarked out of  
1.00

Flag question

Која од следните операции се изведува поефикасно кај двојно поврзани листи отколку кај едностррано поврзани листи?

Select one:

- a. Бришење теме со дадена локација
- b. Вметнување теме после теме со дадена локација
- c. Пребарување зададено теме кај несортирана листа
- d. Изминување на листата за процесирање на секое теме

[Clear my choice](#)

Која од следните операции се изведува поефикасно кај двојно поврзани листи отколку кај едностррано поврзани листи:

- **Бришење теме со дадена локација**

**Question 5**Not yet  
answeredMarked out of  
1.00

Flag question

Динамички поврзана листа е множество...

Select one:

- a. елементи (јазли) што содржат вредност и покажувач кон следен јазел
- b. подредени елементи (јазли) што содржат вредност и покажувач кон следен јазел
- c. подредени елементи (јазли) што содржат вредност и покажувачи кон произволен број јазли
- d. подредени елементи (јазли) што содржат вредност и покажувач кон следната вредност

[Next page](#)[◀ Announcements](#) Jump to... 

Динамички поврзана листа е множество....

- **Б, структура што содржи вредност и покажувач кон следен јазол**

**Question 3**Not yet  
answeredMarked out of  
1.00

Flag question

Замислете дека постојат две решенија на еден проблем. Првото решение има два последователни рекурзивни повика кои го користат скоро почетниот број на елементи. Второто решение ги изминува половина од елементите во циклус и потоа втората половина од елементите во обратен циклус. Кое од решенијата е поефикасно и зошто?

Select one:

- a. Првото, затоа што има логоратимска комплексност за разлика од второто кое има линеарна комплексност.
- b. Второто затоа што го решава проблемот во линеарно време наместо во експоненцијално.
- c. Првото затоа што рекурира е секогаш поефикасна од циклус.
- d. Првото затоа што има експоненцијална комплексност за разлика од второто што има двојна линеарна комплексност.

[Clear my choice](#)

Замислете дека постојат две решенија на еден проблем. Првото решение има два последователни рекурзивни повика кои го користат скоро почетниот број на елементни. Второто решение ги изминува половина од елементите во циклус и потоа втората половина од елементите во обратен циклус. Кое од решенијата е поефикасно?...

- **Второто затоа што го решава проблемот во линеарно време наместо во експоненцијално**

Замислете дека постојат две решенија на еден проблем. Првото решение има рекурзивен повик со за еден намален број на почетни елементи. Второто решение има два вгнездени циклуси со сите елементи. Кое од решениата е поефикасно и зошто?

Select one:

- a. Второто затоа што има квадратна комплексност за разлика од првото кое има експоненцијална комплексност.
- b. Првото затоа што има логаритамска комплексност за разлика од второто што има квадратна комплексност.
- c. Првото, затоа што има линеарна комплексност за разлика од второто кое има квадратна комплексност.
- d. Второто затоа што циклуси се секогаш поефикасна од рекурија.

Замислете дека постојат две решенија на еден проблем. Првото решение има рекурзивен повик со за еден намален број на почетни елементи. Второто решение има два вгнездени циклуси со сите елементи. Кое од решениата е поефикасно и зошто?

- **c . Првото, затоа што има линеарно комплексност за разлика од второто кое има квадратна комплексност.**

Нацртајте ја хеш табела со затворени кофички со должина 9 по вметнување на елементите 10, 35, 18, 2, 19, 26, 9, 28. Како хеш функцијата се користи  $h(x) = x \bmod 9$ . Колку елементи ќе има во кофичките реден број 1 и 2?

- **Int[] niza = new int [9]**

- [0, 1, 2, 3, 4, 5, 6, 7, 8]
- hash(10) = 1
- niza[1] = 10
- hash(35) = 8
- Niza[8] = 35
- hash(26) = 8
- Niza[8] = 35 -> 26
- 0(18, 9) 1(10, 19, 28) 2(2) 3, 4, 5, 6, 7, 8(35, 26)
- 12
- //

0	9	18	
1	28	19	10
2	2		
3			
4			
5			
6			
7			
8	26	35	
9			

**В -точен одговор**

Question 7

Not yet  
answered

Marked out of  
2.00

Flag question

Нека е дадена следната хеш табела која користи отворени кофички (ОВХТ). Хеш функцијата е  $h(x) = x \bmod 9$ , а додека пак чекорот е  $\text{step}(k) = 1$ .

0	1	2	3	4	5	6	7	8
9	18		12	3	14	4	21	

Нека претпоставиме дека се бришат елементите 14 и 21 од хеш табелата, а потоа се повикува внесување на елементите 13 и 23. Во кои кофички ќе бидат сместени елементите 13 и 23?

Select one or more:

- a. 5 и 7
- b. 5 и 8
- c. 8 и 2
- d. 7 и 8

Нека е дадена следната хеш табела која користи отворени кофички ( ОВХТ). Хеш функцијата е  $h(x) = x \bmod 9$ , ... чекорот  $\text{step}(k) = 1$

.... Се бришат елементите 14 и 21 од хеш табелата, а потоа се повикува внесување на елементите 13 и 23 ... во кои кофички сместени 13 и 23 ?

- 5 и 7

**Question 4**Not yet  
answeredMarked out of  
1.00

Flag question

Which algorithm can be used to efficiently calculate a loan installment amount?

Select one:

- a. Merge sort
- b. BFS
- c. Binary search
- d. DFS

Which algorithm efficiently calculate a loan installment amount?

- **Binary search**

**Question 1**Not yet  
answeredMarked out of  
2.00

Flag question

Da se напише функцијата за бројот на извршени операции за дадениот псевдо код и да се пресмета асимптотската горна граница "големо О" за истиот код:

```
simple_computer(x, n)
{
    int sum=0;
    for (int i=0; i<n; i++)
        sum+=2*i*x[i];
    return sum;
}
```

Select one:

- a.  $f(n)=2n^2+4$ ,  $O(n^2)$
- b.  $f(n)=5n+8$ ,  $O(n)$
- c.  $f(n)=3n+6$ ,  $O(3n)$
- d.  $f(n)=2n+8$ ,  $O(n)$

Da se напише функцијата за бројот на извршени операции за дадениот псевдо код и да се пресмета асимптотската горна граница "Големо О" за истиот код:

- **$2n + 8$  или  $5n + 8$  ?  $5n+8$ ,  $O(n)$  ако не се лажам <-**
- **int sum=0; ----- 1**
- **for(int i=0; i<n; i++) ----- 1+(n+1)+n=2n+2**
- **sum+=2\*i\*x[i]; ----- n\*(1+1+1+1)=5n**
- **return sum ----- 1**
- 
- **$1+2n+2+5n+1=7n+4$**

Потребно е да имплементирате хеш табела за складирање на 1000 податоци за студенти, чии клучеви се во форма ХХYZZZZ, каде ХХ е година на запишување (последни 2 цифри), Y е насока - една од 7те и ZZZZ е реден број во рамките на насоката и годината. Која опција за хеширање најмногу би одговарала?

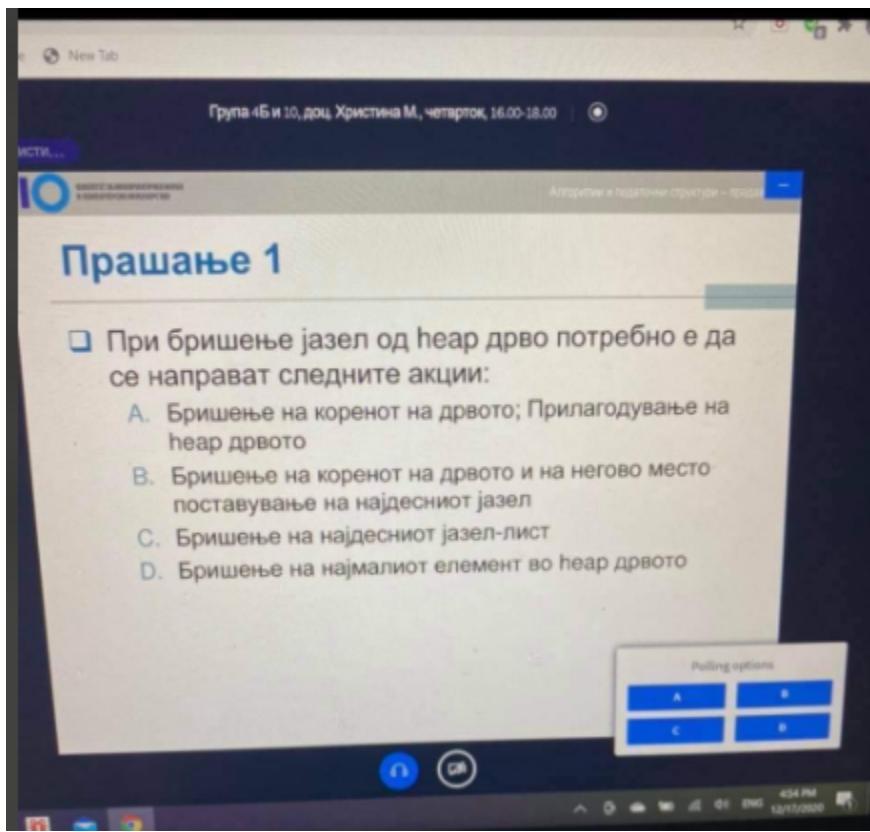
Select one:

- a. 14999 кофички, хеш функција која ги користи сите цифри при пресметка на клучот
- b. 1199 кофички, хеш функција која ги користи првите 4 цифри за пресметка на клучот
- c. 9997 кофички, хеш функција која ги користи првите 2 и последните 2 цифри за пресметка на клучот
- d. 1499 кофички, хеш функција која ги користи последните 4 цифри во пресметка на клучот

[Clear my choice](#)

Потребно е да имплементирате хеш табела за складирање на 1000 податоци, чии клучеви се во форма ХХYZZZZ ХХ е година ... Y е насока и еден од 7-те ZZZZ е реден број..

- **D. 1499 кофички, хеш функција која ги користи последните 4 цифри во пресметка на клучот**



При бришење јазел од heap дрво потребно е да се направат следните акции:

- A. **Бришење на коренот дрвото, прилагодување на heap дрвото.**

3. Што од наведеното е точно?

Select one:

- a. Бројот на нетреминални кај комплетно дрво со степен три не е поврзан со бројот на терминални јазли кај дрва со степен три.
- b. Бројот на нетреминални јазли кај комплетно дрво со степен три е секогаш помал од бројот на терминални јазли.
- c. Бројот на нетреминални јазли кај комплетно дрво со степен три е секогаш поголем или еднаков од бројот на терминални јазли.
- d. Бројот на нетреминални јазли кај комплетно дрво со степен три е секогаш еднаков на бројот на терминални јазли.

Што од наведеното е точно?

**В . Бројот на нетреминални јазли кај комплетно дрво со степен три е секогаш помал од бројот на терминални јазли**

Question 8  
Not yet  
answered

Marked out of  
1.00  


Со која од наведените структури може лесно да се имплементира пронаоѓање на најблиска точка од интерес до тековната локација на мапа?

Select one:

- a. Binary tree
- b. A\* tree
- c. Quad tree
- d. B\* tree

Со која од наведените структури може лесно да се имплементира пронаоѓање на најблиска точка од интерес до тековната локација на мапа?

- Quad tree

**Question 1**Not yet  
answeredMarked out of  
1.00

Flag question

Која е просторната (мемориската) сложеност за бришење на поврзана листа?

Select one:

- a.  $O(n)$
- b. Или  $O(1)$  или  $O(n)$
- c.  $O(1)$
- d.  $O(\log n)$

Која е просторната (мемориската) сложеност за бришење на поврзана листа?

- **$O(1)$**

**Question 2**Not yet  
answeredMarked out of  
1.00

Flag question

Која е комплексноста за внесување на елемент во приоритетна редица во најлош случај?

Select one:

- a.  $O(n^2)$
- b.  $O(n)$
- c.  $O(n * \log n)$
- d.  $O(\log n)$

Која е комплексноста за внесување на елемент во приоритетна редица во најлош случај?

- **$O(n)$**

Доколку дрвото на сликата се трансформира во бинарно дрво тогаш кој јазол ќе биде десното дете на јазолот 5?

**-нема јазол**

**Question 2**

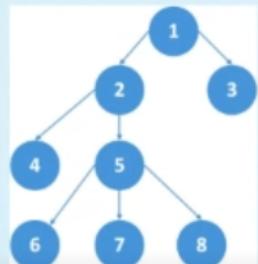
Incorrect

Mark 0.00 out  
of 1.00

Flag  
question

Edit  
question

Koja е висината на јазелот 1 на дрвото на сликата?



Select one:

- a. Нема висина
- b. 1 ✕
- c. 3
- d. 2

Your answer is incorrect.

The correct answer is: 3

Koja е висината на јазелот 1 на дрвото на сликата?

- 3

Flag question

b. наоѓа најбрзо решение за одреден тип на проблем

c. наоѓа најбрзо решение за било кој тип на проблем

d. секогаш наоѓа точно решение за било кој тип на проблем

[Clear my choice](#)

Question 2  
Not yet answered  
Marked out of 1.00  
[Flag question](#)

Koj e степенот на дрвото на сликата?

```
graph TD; 1((1)) --> 2((2)); 1 --> 3((3)); 2 --> 4((4)); 2 --> 5((5)); 5 --> 6((6)); 5 --> 7((7)); 5 --> 8((8));
```

Select one:

a. 3

b. 1

c. 2

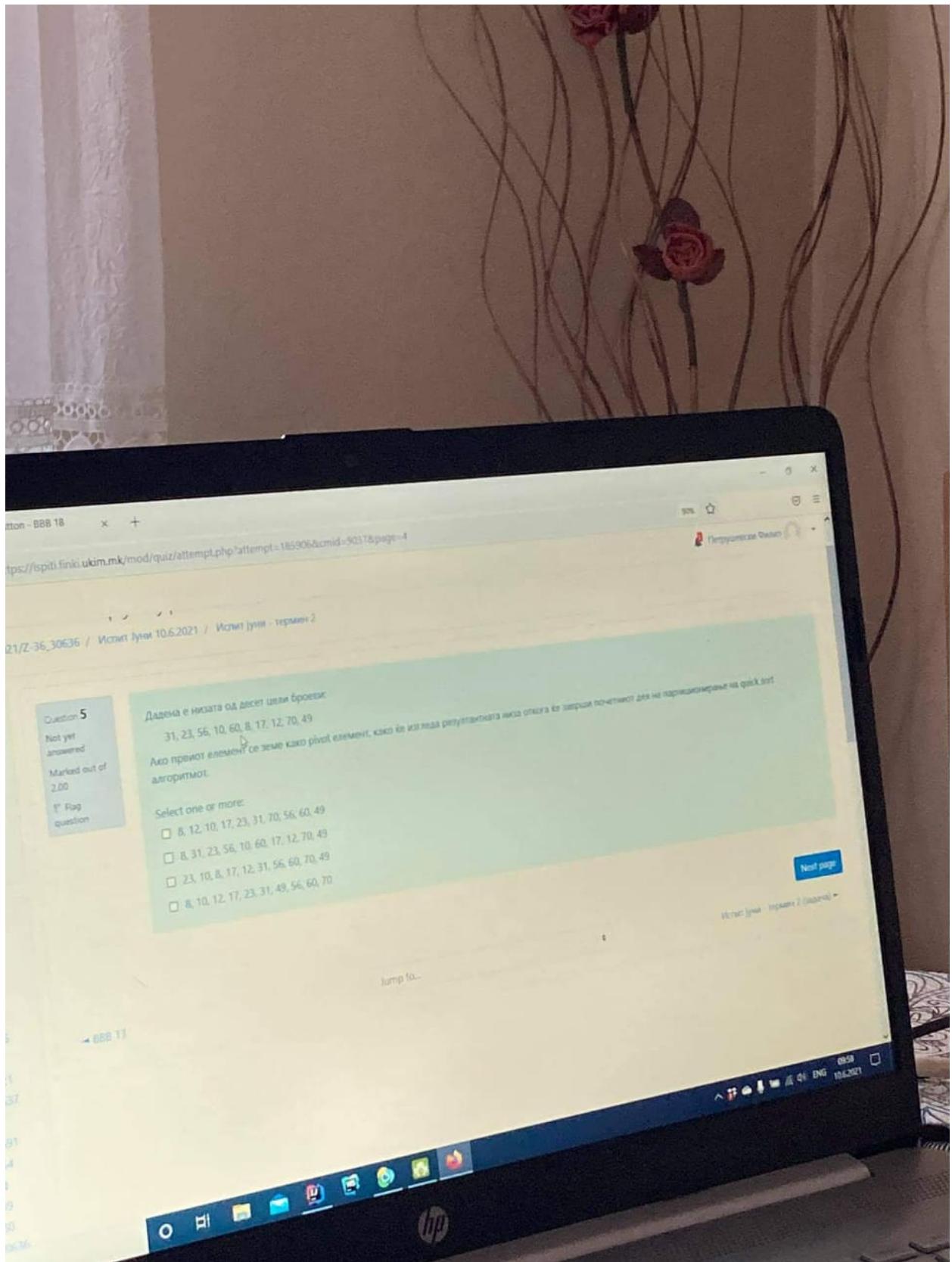
d. Нема степен

[Clear my choice](#)

LG

Koj e степенот на дрвото на сликата?

- 3 (  $5 > 6,7,8$  )



t

- C //bro how? Te taka

Дадени се следните функции:

**push()** : push (додади) елемент на стекот

**pop()** : pop (избриши) го елементот од врвот на стекот

**top()** : врати го елементот кој е на врвот на стекот

Кој ќе биде излезот откако ќе се изврши следната низа на операции:

**push(20);**

**push(4);**

**top();**

**pop();**

**pop();**

**push(5);**

**top();**

Select one:

- a. 5
- b. 4
- c. 20
- d. stack underflow

Дадени се сл функции

- 5

Колкава е сложеноста на алгоритмот со кој се прави спојување на две сортирани листи со големина  $m$  и  $n$  во сортирана листа со големина  $m + n$

- $O(m + n)$

Кога алгоритмот Quick sort има најлоши перформанси:

Select one:

- a. Кога броевите во низата се рамномерно дистрибуирани во одреден опсег
- b. Кога имаме мал број на различни елементи
- c. Кога имаме голем број на различни елементи
- d. Кога низата е скоро сортирана

Кога алгоритмот Quick sort има најлоши перформанси:

- **Кога низата е скоро сортирана**

Question 8  
Correct  
Mark 2.00 out of 2.00  
Flag question  
Edit question

Нека е дадена следната низа од елементи:  
20, 47, 15, 8, 9, 4, 40, 30, 12, 17

Ако се применето merge sort алгоритмот за сортирање на оваа низа, како ќе изгледа низата после второто изминување (second pass) од алгоритмот?

Select one or more:

- 20, 47, 8, 15, 4, 9, 30, 40, 12, 17
- 8, 15, 20, 47, 4, 9, 30, 40, 12, 17 ✓
- 8, 9, 15, 20, 47, 4, 12, 17, 30, 40
- 4, 8, 9, 15, 20, 47, 12, 17, 30, 40
- 15, 20, 47, 4, 8, 9, 12, 30, 40, 17

Нека е дадена следната низа од елементи 20,47,15,8,9,4,40,30,12,17

- **8,15,20,47,4,9,30,40,12,17**

Question 4  
Incorrect  
Mark 0.00 out of 1.00  
Flag question  
Edit question

Нека е дадена репрезентацијата на графот со матрица на соседство  
 $A = [0\ 1\ 0\ 0; 1\ 0\ 0\ 1; 1\ 0\ 0\ 1; 0\ 1\ 1\ 0]$ ,  
каде што редоследот на колоните и редиците е 1, 2, 3, 4, соодветно. Ако графот се претстави преку листа на соседство, тогаш што од понуденото е точно?

Select one:

- a. 4)  $\rightarrow 1 \rightarrow 2 \rightarrow 3$
- b. 1)  $\rightarrow 2 \rightarrow 3$
- c. 3)  $\rightarrow 1 \rightarrow 2 \rightarrow 4$  ✗
- d. 2)  $\rightarrow 1 \rightarrow 4$

The correct answer is: 2)  $\rightarrow 1 \rightarrow 4$

Нека е дадена репрезентација на графот со матрица на соседство

$$A = [0]$$

- D.  $2) \rightarrow 1 \rightarrow 4$

Question 5  
Incorrect  
Mark 0.00 out of 1.00  
[Flag question](#)  
[Edit question](#)

После операцијата вметнување елемент AVL дрвото се дебалансира. Која операција треба да се преземе за да стане правилно AVL дрво.



The diagram shows an AVL tree with root 41. The tree structure is as follows:

```
graph TD; 41 --> 20; 41 --> 65; 20 --> 11; 20 --> 29; 29 --> 30; 29 --> 52; 65 --> 60; 65 --> 91; 91 --> 72; 91 --> 99;
```

Select one:

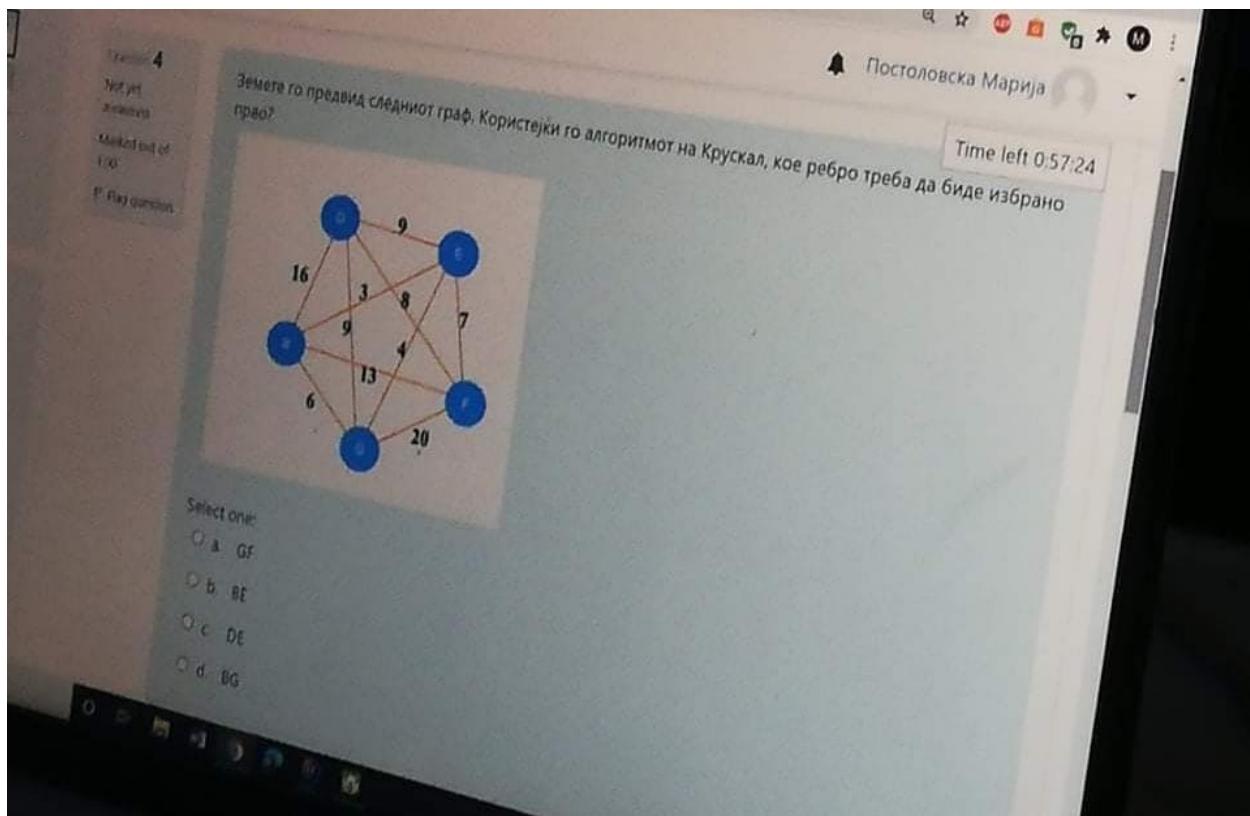
- Единечна ротација - десно
- Двојна ротација - десно, лево
- Двојна ротација - лево, десно
- Единечна ротација - лево X

Your answer is incorrect.  
The correct answer is: Двојна ротација - десно, лево

Балансирано Бинарно пребарувачко(лево помалиот) дрво.

После операцијата вметнување елемент AVL дрвото се дебалансирало.

- **Двојна ротација: - десно, лево**



Крускал

- земи го најмалото ( 3 ) => B.

**Question 8**  
Not yet answered  
Marked out of 2.00  
Flag question

Дадени предмети се претставени како подредени парови (тежина, вредност):  $\{(10,50),(7,28),(4,20),(2,24)\}$ . Капацитетот на ранецот е 13. Да се најде максималната вредност пропоставувајќи дека има само еден од секој предмет и предметите не се деливи (0/1 knapsack).

Select one:

- a. 48
- b. 74
- c. 52
- d. 72

Дадени предмети се претставени како подредени парови ( тежина, вредност):  $\{(10,50),(7,28),(4,20),(2,24)\}$  Капацитетот на ранецот е 13. Да се најде максимална вредност .... Не се деливи 0/1 knapsack).

RANECOT E 13

$$(10, 50) + (2, 24) = (12, 74)$$

$$(7, 28) + (4, 20) + (2, 24) = (13, 72)$$

- 74 е поголемо => B.

Question 8

Not yet  
answered

Marked out of  
2.00

Flag question

Која е сложеноста на имплементација на операцијата додавање, а која на вадење елемент од приоритетна редица имплементирана со подредена низа?

Select one:

- a.  $O(1) - O(1)$
- b.  $O(N) - O(1)$
- c.  $O(N) - O(N)$
- d.  $O(1) - O(N)$

Која сложеноста на имплементација на операцијата додавање, а која на вадење елемент од приоритетна редица имплементирана со подредена низа?

b.  $O(N) - O(1)$

## Подредена низа

$O(n)$

$O(1)$

Question 7

Not yet  
answered

Marked out of  
2.00

Flag question

Дадени предмети се претставени како подредени парови {тежина, вредност}: {{10,60},{7,28},{4,20},{2,24}}. Капацитетот на ранецот е 7. Да се најде максималната вредност препоставувајќи дека има само еден од секој предмет и предметите се деливи (fractional knapsack).

Select one:

- a. 48
- b. 44
- c. 28
- d. 54

Дадени предмети се претставени како подредени парови ( тежина, вредност): ...

Капацитетот на ранецот е 7

**54**

**$24 / 2 = 12$**

**$60 / 10 = 6$**

**$28 / 7 = 4$**

**$20 / 4 = 5$**

**$2 * 12 + 5 * 6 = 54$**

**Deliv e smees da gi delis, deli gi za sekoj weight da vidis koj ti e najvreden.**

**Question 6**Not yet  
answeredMarked out of  
1.00

Flag question

Кој од следниве алгоритми за сортирање има најдобри перформанси кога на влез ќе добие низа што е речиси сортирана (максимално 1 или 2 елементи не се на своите места)?

Select one:

- a. Сортирање со вметнување (insertion sort)
- b. Bubble sort
- c. Брзо сортирање (Quick sort)
- d. Сортирање со спојување (merge sort)

Кој од следниве алгоритми за сортирање има најдобри перформанси кога на влез ќе добие низа што е речиси сортирана( макс 1 или 2 елементи не се на своите места)

**- INSERTION**

Замислете дека постојат две решенија на еден проблем. Првото решение има рекурзивен повик со за еден намален број на почетни елементи?

**- Првото, затоа што има линеарна комплексност- точно е o(n)****Question 5**Not yet  
answeredMarked out of  
1.00

Flag question

Замислете дека постојат две решенија на еден проблем. Првото решение има рекурзивен повик со за еден намален број на почетни елементи. Второто решение има два вгнездени циклуси со сите елементи. Кое од решениата е поефикасно и зашто?

Select one:

- a. Второто затоа што има квадратна комплексност за разлика од првото кое има експоненцијална комплексност.
- b. Првото затоа што има логаритамска комплексност за разлика од второто што има квадратна комплексност.
- c. Првото, затоа што има линеарна комплексност за разлика од второто кое има квадратна комплексност.
- d. Второто затоа што циклуси се секогаш поефикасна од рекурзија.

**Question 4**Not yet  
answeredMarked out of  
1.00

Flag question

За дадена низа  $a[left...right]$ , во кој алгоритам се сретнува следната инструкција (во псевдојазик):

*копирај  $a[x+1...right]$  во  $a[x...right-1]$*

Select one:

- a. Бришење елемент  $a[x]$
- b. Ниту еден од понудените одговори
- c. Додавање елемент  $a[x]$
- d. Копирање една низа во друга

За дадена низа  $a[left...right]$ , во кој алгоритам се сретнува следната инструкција (во псевдојазик):

## - Бришење

Question 3

Not yet  
answered

Marked out of  
1.00

Flag question

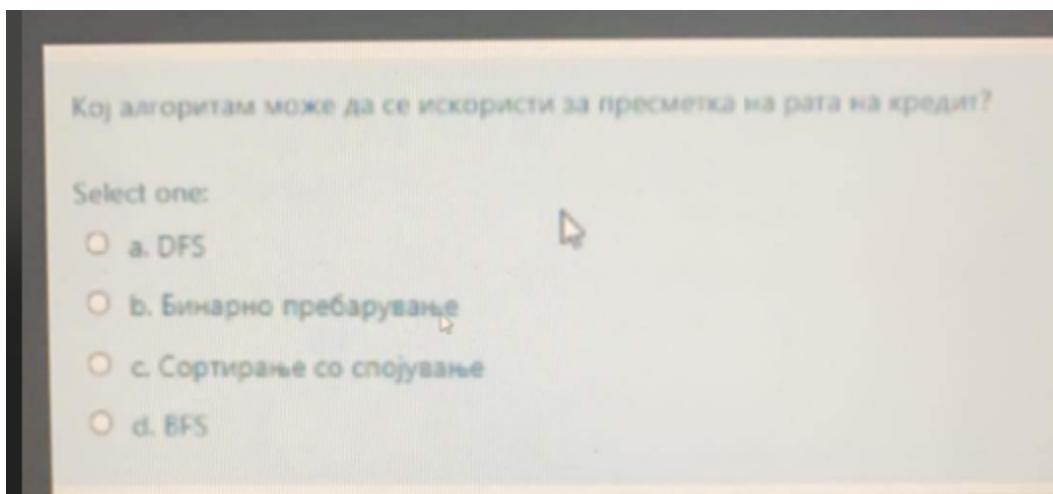
Што од наведеното е точно за стек?

Select one:

- a. Користи приоритет
- b. Имплицитно се имплементира со рекузија
- c. Може да се имплементира само со двојно поврзана листа
- d. Имплицитно се имплементира со итерација

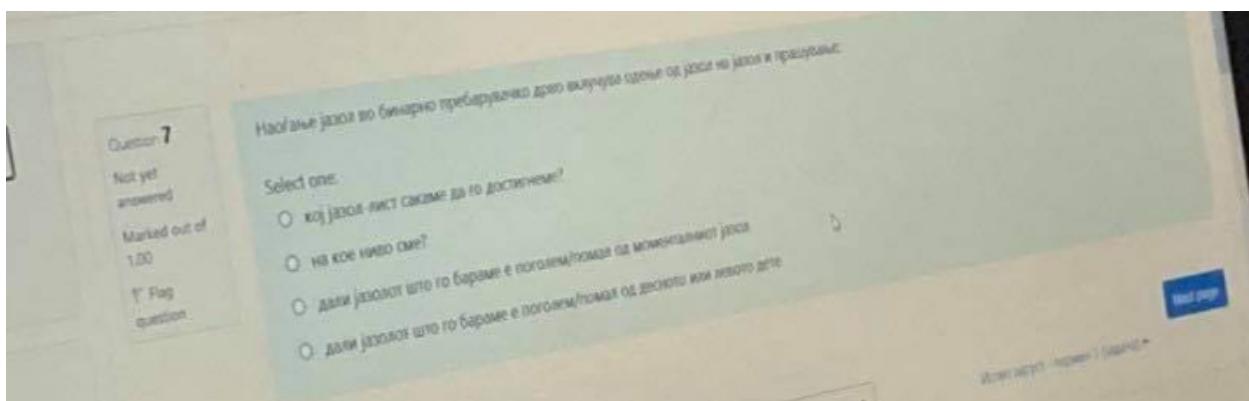
Што од наведното е точно за стек?

- **Имплицитно се имплементира со рекузија**



Кој алгоритам може да се искористи за пресметка на рата на кредит?

**Б. Бинарно преバラување**



Наоѓање јазол во бинарно преバラувачко дрво вклучува

- **дали јазолот што го бараме е поголем/помал од моменталниот јазол.**

Потребно е да имплементирате хеш табела за складирање на 1000 податоци за студенти, чии клучеви се во форма ХХYZZZ, каде ХХ е година на запишување (последни 2 цифри), Y е насока - една од 7те и ZZZZ е реден број во рамките на насоката и годината. Која опција за хеширање најмногу би одговарала?

Select one:

- a. 14999 кофички, хеш функција која ги користи сите цифри при пресметка на клучот
- b. 1199 кофички, хеш функција која ги користи првите 4 цифри за пресметка на клучот
- c. 9997 кофички, хеш функција која ги користи првите 2 и последните 2 цифри за пресметка на клуч
- d. 1499 кофички, хеш функција која ги користи последните 4 цифри во пресметка на клучот

[Clear my choice](#)

Потребно е да имплементирате хеш табела за сладирање на 1000 податоци за студенти, чии клучеви се во форма ХХYZZZ ...

**D. 1499 кофички, хеш функција која ги користи последните 4 цифри во пресметка на клучот.**

Кога алгоритмот Quick sort има најлоши перформанси:

Select one:

- a. Кога броевите во низата се рамномерно дистрибуирани во одреден опсег
- b. Кога имаме мал број на различни елементи
- c. Кога имаме голем број на различни елементи
- d. Кога низата е скоро сортирана

Кога алгоритмот Quick sort има најлоши перформанси?

- **d. Кога низата е скоро сортирана.**

Клучевите 12, 18, 13, 2, 3, 23, 5 и 15 се внесуваат во иницијално празна хеш табела со должина 10 која користи ОВХТ со хеш функција  $h(k) = k \bmod 10$  со  $\text{step}(k) = k \bmod 5$ . Која е резултантната хеш табела?

0	15
1	
2	12
3	13
4	2
5	5
6	3
7	
8	18
9	23

A.

0	5
1	
2	12
3	3
4	2
5	15
6	13
7	23
8	18
9	

B.

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

C.

0	5
1	
2	12
3	13
4	15
5	2
6	
7	3
8	18
9	23

D.

Select one:

- C
- D
- B
- A

Клучевите 12, 18, 13, 2, 3, 23, 5 и 15 се внесуваат во иницијално празна хеш табела со должина 10 која користи ОВХТ со хеш функција  $h(k) = k \bmod 10$  со  $\text{step}(k) = k \bmod 5$ . Која е резултантната хеш табела.

D, A e ne e D

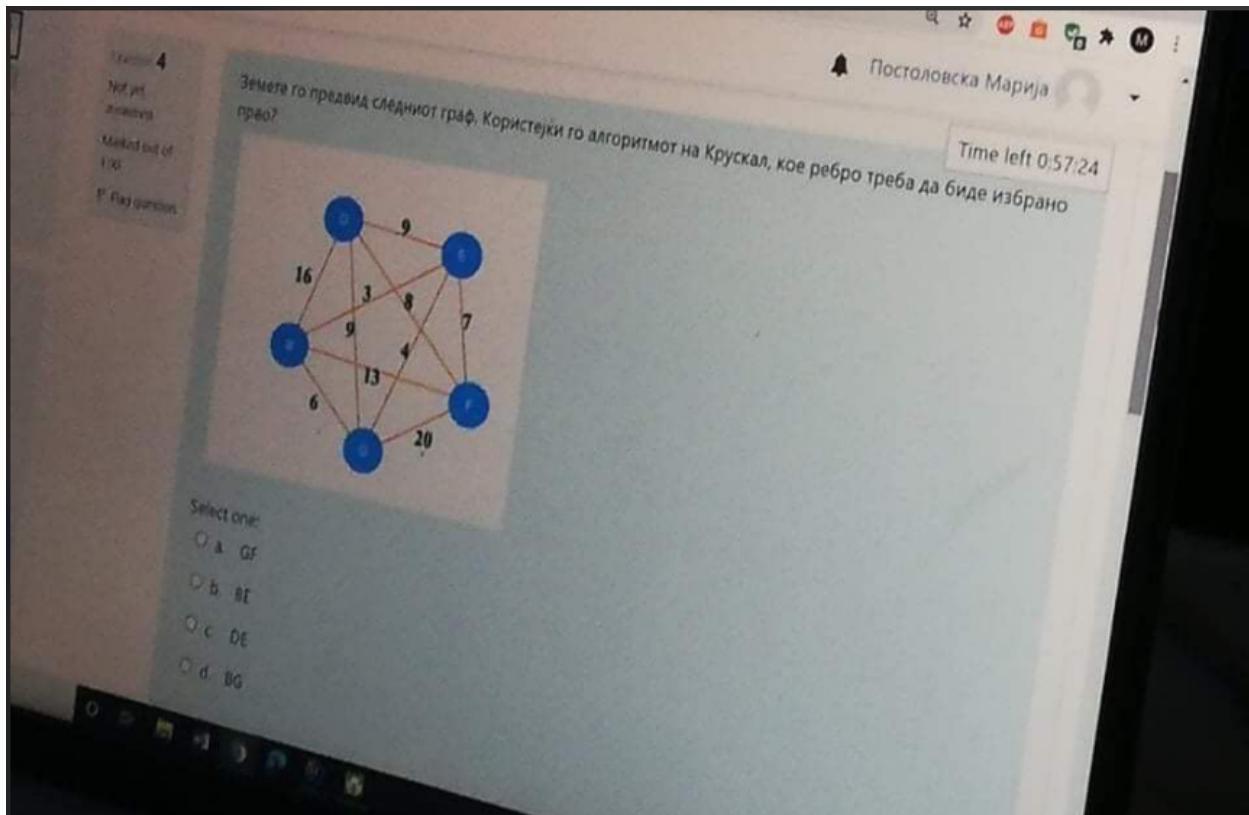
12, 18, 13, 2, 3, 23, 5, 15

$$h(k) = k \bmod 10$$

$$\text{step}(k) = k \bmod 5$$

0	15
1	
2	12
3	13
4	2
5	5
6	3
7	
8	18
9	23

=>A



Земете го предвид следниот граф. Користејќи го алгоритмот на Крускал, кое ребро треба да биде избрано прво?

- **Najmala tezina ( 3 )**

Податочната структура heap, може да се имплементира со помош на еднодимензионална низа поради фактот што:

Select one:

- е комплетно дрво
- е бинарно дрво
- е слабо подредено бинарно дрво
- ги задоволува условите на heap дрво

Податочната структура heap, може да се имплементира со помош на еднодимензионална низа поради фактот што:

- **Е комплетно дрво**

Section 3  
Not yet  
answered  
Marked out of  
1.00  
 Flag question

Time left 1:02:26

Доколку дрвото на сликата се трансформира во бинарно дрво тогаш кој јазол ќе биде десно дете на јазолот 5?

```
graph TD; 1((1)) --> 2((2)); 1 --> 3((3)); 2 --> 4((4)); 5((5)) --> 6((6)); 5 --> 7((7)); 5 --> 8((8))
```

Select one:

- a. Нема да има такво дете
- b. 8
- c. 6
- d. 7

Доколку дрвото на сликата се трансформира бинарното дрво тогаш кој јазол ќе биде десно дете на јазолот 5?

- **Нема да има такво дете**

Time left 1:09:31

Нека е дадена репрезентацијата на графот со матрица на соседство  
 $A = [0 \ 1 \ 0 \ 0; 1 \ 0 \ 0 \ 1; 1 \ 0 \ 0 \ 1; 0 \ 1 \ 1 \ 0]$ ,  
каде што редоследот на колоните и редиците е 1, 2, 3, 4, соодветно. Ако графот се претстави преку листа на соседство, тогаш што од понуденото е точно?

Select one:

- a. 3)  $\rightarrow$  1  $\rightarrow$  2  $\rightarrow$  4
- b. 2)  $\rightarrow$  1  $\rightarrow$  4
- c. 1)  $\rightarrow$  2  $\rightarrow$  3
- d. 4)  $\rightarrow$  1  $\rightarrow$  2  $\rightarrow$  3

Нека е дадена репрезентација на графот со матрица на соседство ...

- **2)  $\rightarrow$  1  $\rightarrow$  4**

Која од следните би била најдобар избор за хеш функција?

Select one:

- a. Нема воопшто колизии, а сложеноста е непозната
- b. Има многу ретки колизии, а сложеноста е  $O(\log N)$
- c. Нема воопшто колизии, а сложеноста е  $O(N)$
- d. Има ретки колизии, а сложеноста е  $O(1)$

Која од следните би била најдобар избор за хеш функција?

- **Има ретки колизии,  $O(1)$**

Нека е дадена репрезентацијата на графот со матрица на соседство  
 $A = [0 \ 1 \ 1 \ 0; 0 \ 0 \ 0 \ 1; 0 \ 0 \ 0 \ 0; 0 \ 0 \ 1 \ 0]$ ,  
каде што редоследот на колоните и редиците е 1, 2, 3, 4, соодветно. Ако се примени пребарување по  
длабочина почнувајќи од темето 2, тогаш кој од следните исписи на темињата е можен.

Time left 1:03:59

Select one:

- a. 1243
- b. 2134
- c. 2314
- d. 243

Ориентиран - С

Jump to

Нека е дадена репрезентација на графот со матрица на соседство ... каде што  
редоследот на колоните и редиците е 1, 2, 3, 4 ..

- **Ориентиран граф е → DFS => 243**

После операцијата вметнување елемент AVL, дрвото се дебалансирало. Која операција треба да се преземе за да стане правилно AVL дрво.



Select one:

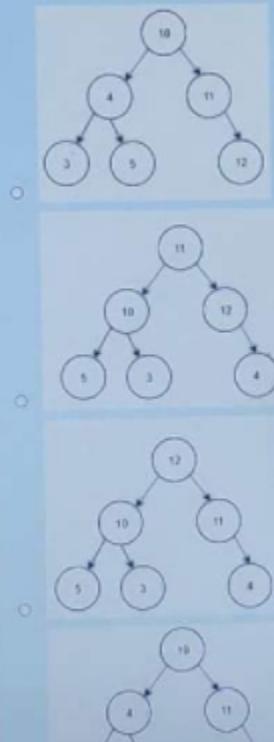
- Единечна ротација - десно
- Двојна ротација - десно, лево
- Двојна ротација - лево, десно
- Единечна ротација - лево x

После операцијата вметнување елемент AVL, дрвото се дебалансирало. Која операција треба да се преземе за да стане правилно AVL дрво.

- **Двојна ротација: десно, лево**

Да се конструира бинарно пребарувачко дрво, ако при изминување со преордер начин се добива следнава секвенца: 10, 4, 3, 5, 11, 12.

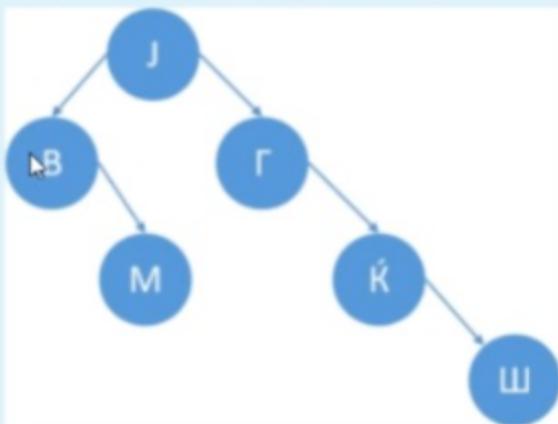
Select one:



Да се конструира бинарно пребарувачко дрво, ако при изминување со ПРЕОРДЕР начин се добива следнава секвенца: 10, 4, 3 ,5, 11, 12:

- **Првото дрво на слика, Корен → лево → десно**

Како ќе изгледа резултатот од inorder изминување на следното дрво:



Select one:

- a. МВШЌГЈ
- b. ЈВМГЌШ
- c. ВМЈГЌШ
- d. МВЈГЌШ

Како ќе изгледа резултатот од inorder изминување на следното дрво:

**с. Лево → Корен → Десно**

Секое В-дрво од ред  $n$ , колку клучеви може да има во внатрешен јазел?

- A.  $n$
- B.  $n-1$
- C. 2
- D. 1

Секое В-дрво од ред  $n$ , колку клучеви може да има во внатрешен јазел?

-  **$n-1$**

Архитектура и компютърни технологии

### Прашање 3

Нека е дадено следното BST.

После бришење на ел. 2 дрвото ќе изгледа вака:

```
graph TD; 6((6)) --- 2((2)); 6 --- 8((8)); 2 --- 1((1)); 2 --- 4((4)); 4 --- 3((3)); 4 --- 5((5))
```

A. 6-2-8-1-4-3-5  
B. 6-1-8-4-3-5  
C. 6-5-8-1-4-3  
D. 6-3-8-1-4-5

Нека е дадено следното BST.

- 6-3-8-1-4-5, иде retardirano rasporedot, на местото на 2ка, треба да иде 3, па иде од ниво по ниво, лево на десно rasporedot

Архитектура и компутерни технологии

### Прашање 2

Бинарно префарувачко дрво е:

- A. Комплетно бинарно дрво во кое секој јазел што претставува корен на свое поддрво има минимална вредност
- B. Полно бинарно дрво во кое елементите во левото поддрво се помали или еднакви на коренот, а елементите во десното поддрво се поголеми од коренот
- C. Бинарно дрво во кое елементите во левото поддрво се строго помали, а оние во десното поддрво строго поголеми од коренот

BigBlueButton - Група 46 и 10...

Polling options

A B C

Бинарно префарувачко дрво е:

- C

### Прашање 3

- При пребарување во ОВНТ, алгоритамот застанува кога:
- A. кофичката  $b$  е никогаш-засфатена
  - B. кофичката  $b$  е засфатена
  - C. кофичката  $b$  е претходно-засфатена
  - D. кофичката  $b$  е никогаш-засфатена или кофичката  $b$  е засфатена со еднаков клуч

При пребарување во ОВНТ, алгоритамот застанува кога:

**D, ако е никогаш засфатена не постоел никад тој елемент, т.е. Hashot не бил ставен во hashtableata, ако е еднаков клуч најден е елементот.**

### Прашање 2

- Кога би требало во една хеш табела да сместиме 1000 податоци кои претставуваат имиња на артикли со соодветен бар код, како ќе ја креираме мапата (клуч, вредност)?
- A. (име-артикл, бар-код)
  - B. (бар-код, име-артикл)
  - C. (име-артикл, име-артикл+бар-код)
  - D. (бар-код, име-артикл+бар-код)

Кога би требало во една хеш табела да сместиме 1000 податоци кои претставуваат имиња на артикли со соодветен бар код, како ќе ја креираме мапата (клуч, вредност) ?

- **Бар-код, име-артикл**

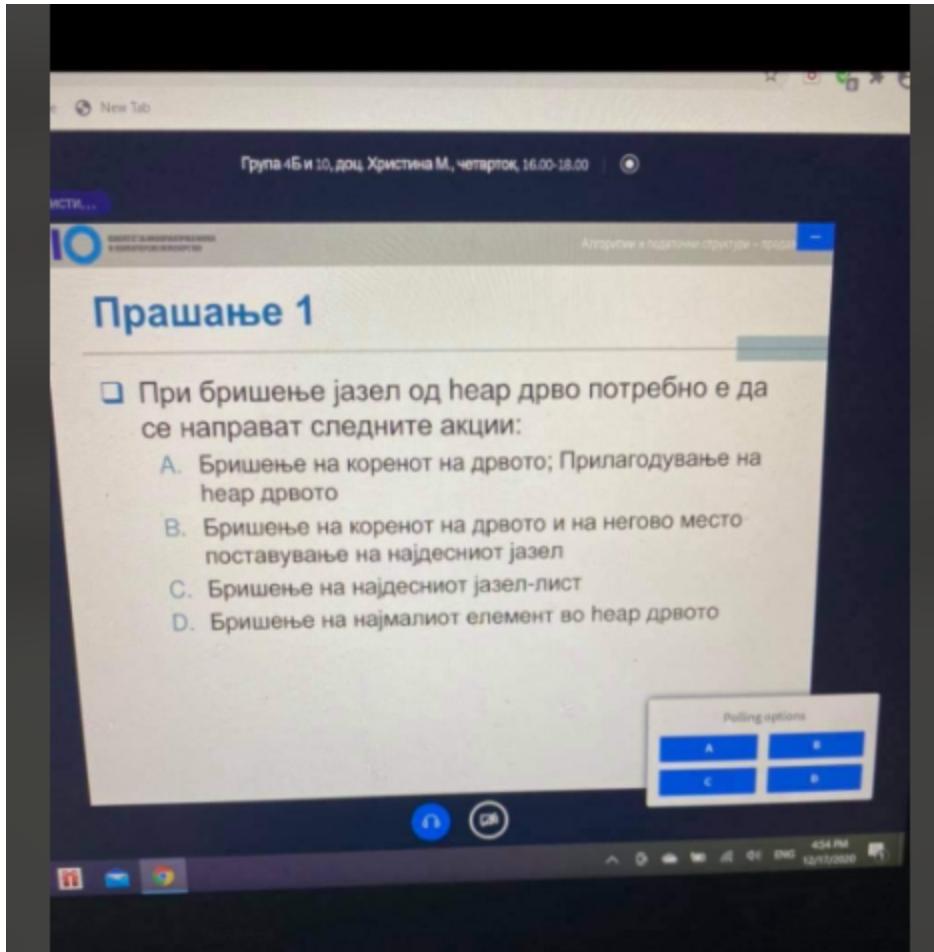
The screenshot shows a poll interface titled "Прашање 1". The question asks: "При работа со СВНТ вметнување елемент во листата на кофичката со клуч hash(key) се прави со која операција од SLL?" with four options: A. insertFirst, B. insertLast, C. insertBefore, and D. insertAfter. Below the question is a red dot. To the right is a "Polling options" box with four blue buttons labeled A, B, C, and D. At the bottom are two circular icons.

При работа со СВНТ вметнување елемент во листата на кофичката со клуч hash(key) се прави со која операција од SLL?

- **insertFirst**

The screenshot shows a poll interface titled "Прашање 4". The question asks: "Најдобри перформанси при користење на ОВНТ се добиваат со:" with four options: A. Една хеш функција и степ функција со чекор 1, B. Една хеш функција и една степ функција со произволен чекор, C. Една хеш функција и степ функција со двојно хеширање, and D. Една хеш функција. Below the question is a red dot. To the right is a "Polling options" box with four blue buttons labeled A, B, C, and D. At the bottom are two circular icons.

Најдобри перформанси при користење на ОВНТ се добиваат со:  
**C. со хеш и степ, двојно хеширање**



При бришење јазел од heap дрво потребно е да се направат следните акции:

A

???

Ako se briske jazel, t.e koren za da se vadi min/max(zavisi dali e min heap ili max)

Togaš treba da se prilagodi, ako se BRISHE (autocorrect me dupi, mora caps) random element od bilo kade u heapot, treba na toa mesto da se stavi najdesniot jazel, PA POSLE TOA PAK SE PRILAGODUVA DRVOTO??????

- The **heap deletion algorithm** in pseudo code:

```

1, Delete a node from the array
    (this creates a "hole" and the tree is no longer "complete")

2. Replace the deletion node
    with the "farthest right node" on the lowest level
    of the Binary Tree
    (This step makes the tree into a "complete binary tree")

3. Heapify (fix the heap):

    if ( value in replacement node < its parent node )
        Filter the replacement node UP the binary tree
    else
        Filter the replacement node DOWN the binary tree
  
```

**Question 2**  
Not yet answered  
Marked out of 1.00  
[Flag question](#)

За најефикасно решавање на проблемот за проверка на добро поставени загради во аритметичките изрази, се користи:

Select one:

- a. стек
- b. приоритетна листа
- c. редица
- d. ниту една од понудените абстрактни податочни структури

За најефикасно решавање за проверка на добро поставени загради во аритметичките изрази, се користи:

**A. стек**

Кое е точното подредување на алгоритмите во однос на нивната сложеност во најдобар случај?

Select one:

- a. Merge sort > Quick sort > Selection sort > Insertion sort
- b. Merge sort > Quick sort > Insertion sort > Selection sort
- c. Insertion sort < Quick sort < Merge sort < Selection sort
- d. Merge sort > Selection sort > Quick sort > Insertion sort

Кое е точното подредување на алгоритмите во однос на нивната сложеност во најдобар случај?

**Best case, insertion > merge  $\Leftrightarrow$  quick > selection(sekad  $n^2$ )**

??? Valjda pod C ama se degeni i sjebale znaci, insertion best case e n

Taka da  $n > n \log n > n \log n > n^2$

Osven ako ne mislat vremenski pa obratnoto da vazhi

## Time Complexities of Sorting Algorithms:

Algorithm	Best	Average	Worst
Quick Sort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$
Bubble Sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$
Merge Sort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$
Insertion Sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$
Selection Sort	$\Omega(n^2)$	$\Theta(n^2)$	$O(n^2)$
Heap Sort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$
Radix Sort	$\Omega(nk)$	$\Theta(nk)$	$O(nk)$
Bucket Sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n^2)$

### Sepak ispagja B?

Time left 0:07:24

Дадени предмети се претставени како подредени парови {тежина, вредност}: {[10,50],[7,28],[4,20],[2,24]}. Капацитетот на ранецот е 13. Да се најде максималната вредност препоставувајќи дека има само еден од секој предмет и предметите не се делливи (0/1 knapsack).

Select one:

- a. 48
- b. 74
- c. 52
- d. 72

### b. 74

Која од следниве карактеристики не е недостаток при користење на низи?

Select one:

- a. Постои можност за неискористеност на мемориски простор ако елементите во низата се помалку отколку алоцираната големина
- b. Додавање на елемент на дадена позиција
- c. Пристап до елементи на специфицирана позиција
- d. Фиксна големина

Која од следниве карактеристики не е недостаток при користење на низи?

Pristap?????

Dodavanje ako shiftas za da ne se preklopuva bi bilo o(N), pristap e sekad o(1)

Замислете дека постојат две решенија на еден проблем. Првото решение има два последователни рекурзивни повика кои го користат скоро почетниот број на елементи. Второто решение ги изминува половина од елементите во циклус и потоа втората половина од елементите во обратен циклус. Кое од решенијата е поефикасно и зошто?

Select one:

- a. Првото затоа што рекурзија е секогаш поефикасна од циклус.
- b. Второто затоа што го решава проблемот во линеарно време наместо во експоненцијално.
- c. Првото затоа што има експоненцијална комплексност за разлика од второто што има двојна линеарна комплексност.
- d. Првото, затоа што има логоратимска комплексност за разлика од второто кое има линеарна комплексност.

Po stoti pat reseno

## Прашање 2

Колкава е комплексноста на раздели-и-владеј решението на проблемот за наоѓање максимален елемент во низа?

- A.  $O(1)$
- B.  $O(n)$
- C.  $O(\log_2 n)$
- D.  $O(n^2)$

Recorded with BigBlueButton

- C

Koj алгоритам за сортирање ќе се изврши за најкратко време во случај кога сите елементи од влезната низа се идентични? Претпоставете ги типичните имплементации на алгоритмите за сортирање.

Select one:

- HEAP SORT
- SELECTION SORT
- INSERTION SORT
- MERGE SORT

### Insertion po stoti pat

- Kolku max teminja moze da ima prost graf?

**N**

$N - 1$

$n^2 - 1$

I uste edno

- Pri prebaruvanje по bfs во матрица на сosedstvo, кој е validen redosled:

1234

4213

Uste 2 opcii