

[Табло](#) / [Моите курсове](#) / [Бакалаври, зимен семестър 2021/2022](#) / [КН](#)

/ [Структури от данни и програмиране \(И, ИС, КН1\), зимен семестър 2021/2022](#) / Зимна сесия 2021/22 г.

/ [Теоретичен изпит \(31.01.2022 г.\)](#)

Започнат на Monday, 31 January 2022, 13:00

Състояние Завършен

Приключен на Monday, 31 January 2022, 14:00

Изминало време 1 час

Оценка 20,27 от 25,00 (81%)

Въпрос **1**

Неправилен отговор

0,00 от максимално 1,00 точки

Ако трябва да намерим най-дългия (като брой дъги) път между два върха в граф, можем ли да използваме за тази цел алгоритъма за обхождане в широчина (breadth-first search)?

Изберете едно:

- ☒ Истина ✖
- ☐ Лъжа

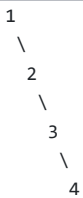
Правилният отговор е "Неистина"

Въпрос **2**

Правилен отговор

1,00 от максимално 1,00 точки

Вярно ли е, че дадено по-долу дърво е двоично дърво за търсене (binary search tree)?



Изберете едно:

- ☒ Истина ✔
- ☐ Лъжа

Правилният отговор е "Истина"

Въпрос **3**

Отговорен

3,00 от максимално 3,00 точки

Обяснете накратко каква е разликата между свързан списък и двоично дърво за търсене. След това посочете какви са предимствата на дървото пред списъка и на списъка пред дървото (ако има такива).

Свързаният списък е линейна структура от данни, а дървото е разклонена. При списъка в общия случай не се изисква никаква специална наредба на елементите, докато при двоично нареденото дърво се изисква -- за всеки връх в дървото трябва да е изпълнено всички елементи в лявото поддърво да са по-малки по ключ от него, а тези отдясно по-големи. Най-голямото предимство на дървото е неговата разклоненост, благодарение на която може да поддържа височината на дървото $O(\log n)$, като имайки и наредбата, това би осигурило логаритмично време за търсене на елемент. Също обаче добавянето и премахването стават логаритмични, а в списъка това не е така. Търсенето при списъка в общия случай е линейно, добавянето в края (ако пазим указател и за края) и добавянето в началото е константно, премахването в началото е константно. Премахването на елемент на произволен индекс е линейно. Кое би било с предимство при избора ни зависи от случая.

Коментар:

Въпрос **4**

Отговорен

2,00 от максимално 2,00 точки

Обяснете накратко какво означава за едно двоично наредено дърво да бъде "изродено".

Обяснете как израждането влияе върху операциите с дървото.

Изродено двоично дърво е такова, чиято височина е станала $O(n)$, където n е броят на елементите. Това, разбира се, забавя очакваните операции с логаритмична сложност по броя върхове, т.е. добавяне, премахване, търсене стават $O(n)$.

Коментар:

Въпрос **5**

Неправилен отговор

0,00 от максимално 1,00 точки

Нека са дадени четирите числа 1, 5, 10 и 100. По колко начина тези числа могат да бъдат наредени в двоично дърво, така че то да бъде двоично дърво за търсене (binary search tree)?

Отговор:

12



Правилният отговор е: 14

Въпрос **6**

Правилен отговор

1,00 от максимално 1,00 точки

Вярно ли е, че всяко двоично наредено дърво за търсене (binary search tree) е също и двоична пирамида (binary heap)?

Изберете едно:

☐ Истина☒ Лъжа ✓

Правилният отговор е "Неистина"

Въпрос 7

Частично правилен отговор

0,60 от максимално 1,00 точки

Нека изпълняваме бързо сортиране (quicksort) върху масив с N -елемента.

Делителния елемент (pivot) за всеки pass избираме да бъде първият елемент в частта от масива, която се сортира.

Кое/кои от следните условия водят до най-лошия случай на изпълнение - $O(N^2)$?

Изберете едно или повече:

- ☒ a. Масивът е (почти) сортиран ✓
- ☐ b. Избран е делителен елемент (pivot), който е прекалено малък (например най-малкият елемент в масива).
- ☒ c. Масивът е (почти) сортиран в обратен ред ✓
- ☐ d. Избран е делителен елемент (pivot), който е прекалено голям (например най-големият елемент в масива).
- ☒ e. Всички (или почти всички) елементи в масива са еднакви ✓

Your answer is partially correct.

Вие правилно сте избрали 3.

Правилните отговори са: Избран е делителен елемент (pivot), който е прекалено малък (например най-малкият елемент в масива), Избран е делителен елемент (pivot), който е прекалено голям (например най-големият елемент в масива), Масивът е (почти) сортиран, Масивът е (почти) сортиран в обратен ред, Всички (или почти всички) елементи в масива са еднакви

Въпрос 8

Правилен отговор

1,00 от максимално 1,00 точки

Нека е дадено произволно двоично наредено дърво за търсене (binary search tree), което съдържа N -елемента.

Каква е сложността на търсенето, вмъкването и премахването на елемент в най-лошия случай?

Изберете едно

- ☐ a. $O(1)$
- ☐ b. $O(\log N)$
- ☒ c. $O(N)$ ✓
- ☐ d. $O(N \log N)$
- ☐ e. $O(N^2)$
- ☐ f. $O(2^N)$

Your answer is correct.

Правилният отговор е: $O(N)$

Въпрос 9

Неправилен отговор

0,00 от максимално 1,00 точки

По-долу е даден кодът на функцията Mystery, която сортира масив от цели числа. Кой алгоритъм използва функцията?

```
void Mystery(int* pArr, size_t Size)
{
    size_t i = Size / 2;

    while (i--)
        Help(pArr, i, Size);

    i = Size;

    while (--i)
    {
        std::swap(pArr[0], pArr[i]);
        Help(pArr, 0, i);
    }
}

void Help(int* pArr, size_t pos, size_t Size)
{
    int elem(pArr[pos]);

    size_t ni = pos;
    size_t si = pos * 2 + 1;

    while (si < Size)
    {
        if (si < Size - 1 && pArr[si] < pArr[si + 1])
            si++;

        if (elem > pArr[si])
            break;

        pArr[ni] = pArr[si];
        ni = si;
        si = si * 2 + 1;
    }

    pArr[ni] = elem;
}
```

Изберете едно

- ☐ a. Метод на мехурчето (Bubble sort)
- ☐ b. Сортировка на Шел (Shell sort)
- ☐ c. Сортиране чрез клатене (Shaker sort)
- ☐ d. Сортиране чрез сливане (Merge sort)
- ☒ e. Функцията не реализира никой от алгоритмите посочени в другите отговори
- ☐ f. Пирамидално сортиране (Heap sort)
- ☐ g. Броене на честоти (Counting sort)



Your answer is incorrect.

Правилният отговор е: Пирамидално сортиране (Heap sort)

Въпрос **10**

Правилен отговор

1,00 от максимално 1,00 точки

Нека е даден **сортиран** масив, който съдържа N -елемента.

Каква ще бъде сложността на всяка от дадените по-долу операции с масива?

Забележка: В списъка с отговорите със знак $^$ е обозначена операцията степенуване. По-конкретно:

- $O(N^2)$ обозначава сложността $O(N^2)$
- $O(2^N)$ обозначава сложността $O(2^N)$

Намиране на максималния елемент в масива



Проверка дали даден елемент се съдържа в масива (с двоично търсене)



Добавяне на елемент на първа позиция в масива



Вмъкване на елемент на произволна позиция



Проверка дали даден елемент се съдържа в масива (с изчерпващо търсене)



Извличане на стойността на елемент намиращ се на даден индекс k в масива



Your answer is correct.

Правилният отговор е: Намиране на максималния елемент в масива $\rightarrow O(1)$, Проверка дали даден елемент се съдържа в масива (с двоично търсене) $\rightarrow O(\log(N))$, Добавяне на елемент на първа позиция в масива $\rightarrow O(N)$, Вмъкване на елемент на произволна позиция $\rightarrow O(N)$, Проверка дали даден елемент се съдържа в масива (с изчерпващо търсене) $\rightarrow O(N)$, Извличане на стойността на елемент намиращ се на даден индекс k в масива $\rightarrow O(1)$

Въпрос 11

Правилен отговор

1,00 от максимално 1,00 точки

Нека е даден **едносвързан** списък, който съдържа N -елемента. Считаме, че представянето е такова, че разполагаме с указатели към първата и последната кутия в списъка.

Каква ще бъде сложността на всяка от дадените по-долу операции със списъка?

Забележка: В отговорите със знак $^$ е обозначена операцията степенуване. По-конкретно:

- $O(N^2)$ обозначава сложността $O(N^2)$
- $O(2^N)$ обозначава сложността $O(2^N)$

Вмъкване на елемент на произволна позиция в списъка	<input type="text" value="O(N)"/>	✓
Изтриване на последния елемент на списъка	<input type="text" value="O(N)"/>	✓
Проверка дали даден елемент се съдържа в списъка	<input type="text" value="O(N)"/>	✓
Вмъкване на елемент на последна позиция в списъка	<input type="text" value="O(1)"/>	✓
Изтриване на първия елемент на списъка	<input type="text" value="O(1)"/>	✓
Вмъкване на елемент на първа позиция в списъка	<input type="text" value="O(1)"/>	✓
Намиране на максималния елемент в списъка	<input type="text" value="O(N)"/>	✓

Your answer is correct.

Правилният отговор е: Вмъкване на елемент на произволна позиция в списъка $\rightarrow O(N)$, Изтриване на последния елемент на списъка $\rightarrow O(N)$, Проверка дали даден елемент се съдържа в списъка $\rightarrow O(N)$, Вмъкване на елемент на последна позиция в списъка $\rightarrow O(1)$, Изтриване на първия елемент на списъка $\rightarrow O(1)$, Вмъкване на елемент на първа позиция в списъка $\rightarrow O(1)$, Намиране на максималния елемент в списъка $\rightarrow O(N)$

Въпрос 12

Неправилен отговор

0,00 от максимално 1,00 точки

Нека имаме имплементация на separate chaining или linear probing хеш съдържащ N елемента. Ако трябва да намерим най-малкия или най-големия елемент в нея, това в общия случай е операция със сложност:

Изберете едно

- ☐ a. $O(\log N)$
- ☐ b. $O(N^2)$
- ☐ c. $O(N)$
- ☐ d. $O(2^N)$
- ☒ e. $\backslash(O(1))\backslash$



Your answer is incorrect.

Правилният отговор е: $\backslash(O(N))\backslash$

Въпрос 13

Правилен отговор

1,00 от максимално 1,00 точки

Вярно ли е, че алгоритъмът за обхождане в дълбочина (depth-first search) винаги има нужда от повече памет за работата си, отколкото алгоритъмът за обхождане в широчина (breadth-first search)?

Изберете едно:

- ☐ Истина
- ☒ Лъжа ✓

Правилният отговор е "Неистина"

Въпрос 14

Правилен отговор

1,00 от максимално 1,00 точки

Трябва да създадем контейнер, за който имаме следните изисквания (N е броят на елементите, които се съдържат в контейнера):

1. Намирането на k -ия най-малък елемента да става със сложност $O(1)$;
2. Намирането на k -ия най-голям елемента да става със сложност $O(1)$;
3. Търсенето за произволен елемент да става със сложност не по-голяма от $O(\log N)$;

Коя от следните структури от данни може да се използва за реализирането му?

Изберете едно

- ☐ a. Сортиран списък с две връзки
- ☐ b. Двоично наредено дърво
- ☐ c. Пирамида (heap)
- ☐ d. Динамичен масив
- ☒ e. Сортиран динамичен масив



Your answer is correct.

Правилният отговор е: Сортиран динамичен масив

Въпрос 15

Частично правилен отговор

0,67 от максимално 1,00 точки

В общия случай кои от следните са предимства на балансираните двоични наредени дървета пред хешовете?

Изберете едно или повече:

- ☒ a. Ако имаме стойност N и искаме да намерим най-близкия по стойност елемент съхранен в структурата, това става по-бързо в дървото. ✓
- ☐ b. Търсенето на елемент става по-бързо в дървото.
- ☒ c. Ако искаме да обходим съхранените елементите в нарастващ ред, това става по-бързо в дървото. ✓
- ☐ d. Ако искаме да намерим всички елементи от интервала $[A, B]$, които се съдържат в структурата, това става по-бързо в дървото.
- ☐ e. Добавянето на елемент става по-бързо в дървото.
- ☐ f. Всички хешове имат максимален брой елементи N , които могат да се запишат в тях, докато дървото може да нараства колкото поискаме.

Your answer is partially correct.

Вие правилно сте избрали 2.

Правилните отговори са: Ако искаме да обходим съхранените елементите в нарастващ ред, това става по-бързо в дървото., Ако имаме стойност N и искаме да намерим най-близкия по стойност елемент съхранен в структурата, това става по-бързо в дървото., Ако искаме да намерим всички елементи от интервала $[A, B]$, които се съдържат в структурата, това става по-бързо в дървото.

Въпрос **16**

Отговорен

2,00 от максимално 2,00 точки

Нека е даден Separate Chaining хеш, който съхранява цели числа и използва хешираща функция еквивалентна на следното:

```
unsigned int hash(int x) {  
    return abs(x % 10);  
}
```

Дайте пример за такава поредица от числа, която би предизвикала серия от колизии и в резултат всяко добавяне (или търсене) на число от поредицата ще бъде със сложност $O(N)$, вместо с $O(1)$. Тук N е броят на числата съхранени в хеша.

1, 11, 111, 1111, ...

Коментар:

Въпрос **17**

Отговорен

2,00 от максимално 2,00 точки

Обяснете как се отразява Data Locality на операциите със свързан списък.

Обяснете какъв е ефектът върху основните операции и коментирайте дали той е положителен или негативен.

Свързаният списък има доста лоша локалност. Кеш линиите, които биха се заредили, не биха били от голяма помощ при операциите, защото кутийките на списъка за разпръснати из произволни места из паметта, което е негативно.

Коментар:

Въпрос 18

Правилен отговор

1,00 от максимално 1,00 точки

Трябва да създадем контейнер, за който имаме следните изисквания:

1. Да можем да добавяме произволен брой елементи към контейнера;
2. Добавянето да бъде със сложност $O(1)$ (може да бъде амортизирана);
3. Няма нужда елементите да се поддържат в някакъв определен ред

Кои от следните структури от данни може да се използват за реализирането му?

Изберете едно или повече:

- ☒ a. Свързан списък с две връзки
- ☒ b. Динамичен масив
- ☐ c. Двоично наредено дърво
- ☒ d. Хеш
- ☐ e. Пирамида (heap)
- ☒ f. Свързан списък с една връзка
- ☐ g. Статичен масив



Your answer is correct.

Правилните отговори са: Динамичен масив, Свързан списък с една връзка, Свързан списък с две връзки, Хеш

Въпрос 19

Правилен отговор

1,00 от максимално 1,00 точки

Нека е дадено празно двоично наредено дърво за търсене T .

В дървото добавяме числата от 1 до 9 в следния ред: 7, 1, 3, 5, 4, 9, 2, 6, 8.

Ако обхождаме дървото в ред ляво-корен-дясно (in-order) и извеждаме числата на екрана, какво ще се изведе на екрана?

(От падащите менюта изберете стойностите така, че да се получат числата точно в реда, в който ще се изведат на екрана)

1	✓	2	✓	3	✓	4	✓	5	✓	6	✓	7	✓	8	✓	9	✓
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Въпрос **20**

Правилен отговор

1,00 от максимално 1,00 точки

Кое от следните е необходимо, за да може да приложим алгоритъма за двоично търсене (binary search) върху даден масив?

Изберете едно

- ☐ a. Никое от посочените -- алгоритъмът може да се приложи върху произволен масив
- ☒ b. Масивът да е сортиран
- ☐ c. Масивът да има не повече от 2^{32} елемента
- ☐ d. Масивът да не съдържа отрицателни числа
- ☐ e. В масива да няма повтарящи се елементи
- ☐ f. Масивът да съдържа цели числа



Your answer is correct.

Правилният отговор е: Масивът да е сортиран

[◀ Практически изпит \(31.01.2022 г.\)](#)

Отиди на ...

[Разписание за защиты на проекти по СДП ▶](#)