

Навчальна програма

Інформація про курс

Обговорення

Прогрес

Конспект лекцій

Будь ласка, зверніть увагу! Фінальний іспит оцінюється в 30 балів, які враховуються для отримання сертифікату.

Іспит складається з 30 тестових питань. Ви відповідаєте на кожне питання окремо. Зверніть особливу увагу на те, що у вас є лише одна спроба відповіді для кожного питання. Тому будьте максимально уважні!

Вкінці після всіх запитань ви можете знайти посилання на обговорення у форумі питань з іспиту. Якщо ви не впевнені у формулюванні питання або вважаєте, що ваша відповідь коректна, то звертайтеся на форум.

ПИТАННЯ 1 (1/1 бал)

Згадайте процедуру Partition методу швидкого сортування. Припустимо, що наступний масив був щойно розбитий відповідно до цієї процедури по відношенню до деякого опорного елемента: [3, 1, 2, 4, 5, 8, 7, 6, 9]. І цей опорний елемент вже зайняв своє відповідне місце в масиві.

Який з цих елементів міг бути обраним в якості опорного? Оберіть всі можливі варіанти.

☐ 2☐ 3☒ 4☒ 5☒ 9


Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 2 (1/1 бал)

Вам задано k відсортованих масивів, кожний з яких містить n елементів, і ви хочете об'єднати їх в один масив з kn елементів. Розглянемо наступний підхід. Розділимо k масивів на $k/2$ пар масивів та використаємо процедуру злиття Merge алгоритму сортування злиттям. Тепер у вас залишиться $k/2$ відсортованих масивів, кожний з яких буде містити $2n$ елементів. Будемо повторювати ці кроки до тих пір, поки не отримаємо один масив з kn елементів.

Який час роботи цієї процедури як функції k і n ?


- ☒ $\Theta(nk \log k)$ 
- ☐ $\Theta(nk^2)$
- ☐ $\Theta(n \log k)$
- ☐ $\Theta(nk \log n)$

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 3 (1/1 бал)

Як виглядає рекурентне рівняння для найгіршого випадку процедури швидкого сортування QuickSort та яка часова складність в цьому випадку?

- ☐ $T(n) = T(n - 2) + \Theta(n)$ і часова складність $T(n) = \Theta(n^2)$
- ☒ $T(n) = T(n - 1) + \Theta(n)$ і часова складність $T(n) = \Theta(n^2)$ 

☐ $T(n) = 2T(n/2) + \Theta(n)$ і часова складність $T(n) = \Theta(n \log n)$

☐ $T(n) = T(n/10) + T(9n/10) + \Theta(n)$ і часова складність $T(n) = \Theta(n \log n)$

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 4 (1/1 бал)

Розглянемо функцію

```
Function(ціле число n):  
    count = 0  
    i = n  
    while i>0:  
        for j від 0 до i-1:  
            count = count + 1  
        i = i/2  
    return count
```

Яка часова складність цієї функції?

☐ $\Theta(n^2)$

☐ $\Theta(n \log n)$

☒ $\Theta(n)$ ✓

☐ $\Theta(n \log n \log n)$

ПОЯСНЕННЯ ВІДПОВІДІ

Якщо розписати суму кількостей ітерацій для зовнішнього циклу, то отримаємо: $n + n/2 + n/4 + \dots + 1$. Це є сума геометричної прогресії:

$$n/2^0 + n/2^1 + n/2^2 + \dots + n/2^{\log_2 n} = n \frac{1 - 1/2^{\log_2 n + 1}}{1 - 1/2} = n \frac{1 - 1/2n}{1/2} = 2n - 1 = \Theta(n)$$

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 5 (1/1 бал)

Позначимо через $W(n)$ та $A(n)$ час роботи деякого алгоритму в найгіршому та середньому випадках відповідно, коли розмірність вхідних даних становить n .

Яке з наступних співвідношень завжди є правильним?

☐ $A(n) = \Omega(W(n))$

☐ $A(n) = \Theta(W(n))$

☒ $A(n) = O(W(n))$



ПОЯСНЕННЯ ВІДПОВІДІ

Час роботи алгоритму в найгіршому випадку завжди є більшим або рівним середньому часу роботи.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 6 (1/1 бал)

В деякому змаганні розглядається чотири функції. Кожна з функцій має один цикл for і всередині цього циклу кожна функція має один й той самий набір інструкцій.

Нижче наведені чотири варіанти циклів for. Для вхідних даних розмірності n (додатне число), яка з функцій буде більш ефективною? Зміст самої задачі, яку розв'язують функції, не є важливим.

- ☐ `for(i = 0; i < n; i = i + 1)`
- ☐ `for(i = 0; i < n; i = i + 2)`
- ☒ `for(i = 1; i < n; i = i * 2)` ✓
- ☐ `for(i = n; i > -1; i = i / 2)`

ПОЯСНЕННЯ ВІДПОВІДІ

Часова складність першого циклу - $\Theta(n)$. Часова складність другого циклу - $\Theta(n/2) = \Theta(n)$. Часова складність третього циклу - $\Theta(\log n)$. Четвертий цикл ніколи не завершиться.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 7 (1/1 бал)

Оберіть ті твердження, які є правильними.

- ☒ $(n + k)^m = \Theta(n^m)$, де k і m - константи ✓
- ☒ $2^{n+1} = \Theta(2^n)$ ✓
- ☐ $2^{2n+1} = \Theta(2^n)$


ПИТАННЯ 8 (1/1 бал)

Розглянемо наступні дві функції.

```
Fun1(ціле число n):  
    if n ≤ 1 return n  
    else return 2 * Fun1(n - 1)
```

```
Fun2(ціле число n):  
    if n ≤ 1 return n  
    else return Fun2(n-1) + Fun2(n-1)
```

Яка часова складність цих двох функцій?

- ☐ $\Theta(2^n)$ для обох функцій
- ☒ $\Theta(n)$ для Fun1 та $\Theta(2^n)$ для Fun2 
- ☐ $\Theta(2^n)$ для Fun1 та $\Theta(n)$ для Fun2
- ☐ $\Theta(n)$ для обох функцій

ПОЯСНЕННЯ ВІДПОВІДІ

Часова складність для Fun1 може бути записана як $T(n) = T(n - 1) + C$, звідки $T(n) = \Theta(n)$. Часова складність для Fun2 може бути записана як $T(n) = 2T(n - 1) + C$, звідки $T(n) = \Theta(2^n)$.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 9 (1/1 бал)

Невідсортований масив містить n різних елементів. Як оцінюється кількість порівнянь у випадку пошуку будь-якого елементу в масиві, який не є ані мінімальним, ані максимальним?

- ☐ $\Theta(n \log n)$
- ☐ $\Theta(n)$
- ☐ $\Theta(\log n)$
- ☒ $\Theta(1)$ ✓

ПОЯСНЕННЯ ВІДПОВІДІ

Для пошуку такого елементу достатньо обрати перші три елементи масиву та порівняти їх між собою. Той, який буде ані мінімальним, ані максимальним серед трьох, і є шуканий елемент. Час його пошуку не залежить від довжини масиву і становить $\Theta(1)$.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 10 (1/1 бал)

Розглянемо ситуацію, коли у вас немає функції для обрахунку степеню (наприклад, аналог функції `pow()` у мові C) і вам потрібно обрахувати x^n , де x - довільне число і n - ціле додатне число. Яким може бути найкращий час роботи вашої степеневої функції?

- ☐ $\Theta(n)$

- ☐ $\Theta(n \log n)$
- ☒ $\Theta(\log n)$ ✓
- ☐ $\Theta(\log(\log n))$

ПОЯСНЕННЯ ВІДПОВІДІ

Наведена нижче функція працює за принципом декомпозиції. Її рекурентне співвідношення можна представити як $T(n) = T(n/2) + \Theta(1)$, розв'язок якого становить $T(n) = \Theta(\log n)$.

```
Power(x, y):  
    if y = 0:  
        return 1  
    temp = Power(x, y/2)  
    if y mod 2 = 0:  
        return temp*temp  
    else  
        return x*temp*temp
```

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 11 (1/1 бал)

Розглянемо задачу пошуку підмасиву з максимальною сумою. На вхід подається деякий масив чисел A . Потрібно знайти таку неперервну послідовність чисел в масиві A , що їх сума буде найбільшою. Наприклад, для масиву $[12, -13, -5, 25, -20, 30, 10]$ найбільша сума неперервної послідовності елементів буде становити 45, де сама послідовність - $[25, -20, 30, 10]$.

Очевидний розв'язок цієї задачі полягає в тому, щоб рахувати суми всіх підмасивів починаючи з кожного елемента вхідного масиву та повернути підмасив з максимальною сумою. Проте можна застосувати метод декомпозиції та розробити алгоритм, який буде працювати краще, ніж згаданий вище. Який буде час роботи цього нового алгоритму?

☐ $\Theta(n)$

☒ $\Theta(n \log n)$ ✓

☐ $\Theta(\log n)$

☐ $\Theta(n^2)$

ПОЯСНЕННЯ ВІДПОВІДІ

Для поставленої задачі можна використати наступний алгоритм, який заснований на принципі декомпозиції.

1) Розбити вхідний масив на дві половини

2) Повернути максимальний з наступних трьох випадків:

a) Максимальну суму підмасиву з лівої половини (рекурсивний виклик)

b) Максимальну суму підмасиву з правої половини (рекурсивний виклик)

c) Максимальну суму підмасиву, в якого початок знаходиться в лівій половині, а кінець - в правій.

Для пошуку значення на пункті c) ми можемо скористатись наступною процедурою. Рухатись від середньої точки вліво і шукати підмасив з максимальною сумою. Потім рухатись від середньої точки вправо і також шукати підмасив з максимальною сумою. В кінці додати ці дві суми.

Рекурентне співвідношення становить: $T(n) = 2T(n/2) + \Theta(n)$, а його розв'язок - $T(n) = \Theta(n \log n)$.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 12 (1/1 бал)

Припустимо, ми маємо алгоритм для пошуку медіани в масиві, який працює за час $\Theta(n)$. Розглянемо метод швидкого сортування, в якому спочатку знаходиться медіана масиву із використанням згаданого алгоритму, а потім ця медіана використовується в якості опорного елементу процедури розбиття. Який буде час роботи такого модифікованого алгоритму швидкого сортування в найгіршому випадку?

- ☐ $\Theta(n^2 \log n)$
- ☐ $\Theta(n^2)$
- ☐ $\Theta(n \log n \log n)$
- ☒ $\Theta(n \log n)$ ✓

ПОЯСНЕННЯ ВІДПОВІДІ

Рекурентне співвідношення для такого модифікованого алгоритму має вигляд $T(n) = 2T(n/2) + \Theta(n)$. Його розв'язок - $T(n) = \Theta(n \log n)$.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 13 (1/1 бал)

Розглянемо модифікацію методу сортування злиттям, коли вхідний масив розбивається у співвідношенні 1 до 2, тобто одна частина буде містити третину вхідного масиву, а друга - решти дві третини. Який час роботи цієї модифікації алгоритму в найгіршому випадку?

- ☐ $\Theta(n \log_3 n)$
- ☐ $\Theta(n \log_{2/3} n)$
- ☐ $\Theta(n \log_{1/3} n)$
- ☒ $\Theta(n \log_{3/2} n)$ ✓

ПОЯСНЕННЯ ВІДПОВІДІ

Рекурентне співвідношення для такого модифікованого алгоритму має вигляд $T(n) = T(n/3) + T(2n/3) + \Theta(n)$. Його розв'язок - $T(n) = \Theta(n \log_{3/2} n)$.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 14 (1/1 бал)

Який з наступних алгоритмів буде працювати найменший час, якщо вхідний масив складається тільки з однакових елементів? Припускаємо, що розглядаються стандартні реалізації алгоритмів.

- ☒ Сортування включенням (Insertion Sort) ✓
- ☐ Пірамідальне сортування (Heap Sort)
- ☐ Сортування злиттям (Merge Sort)
- ☐ Швидке сортування (Quick Sort)

ПОЯСНЕННЯ ВІДПОВІДІ


Алгоритм сортування включенням потребує час $\Theta(n)$, коли всі елементи вхідного масиву однакові. Також за час $\Theta(n)$ буде працювати алгоритм Heap Sort, проте приховані константи в цьому випадку будуть більші, ніж в Insertion Sort, через витрати часу на побудову піраміди.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 15 (1/1 бал)

Масив з n символічних рядків, кожний з яких має довжину n , відсортовується в лексикографічному порядку за допомогою методу сортування злиттям. Який час роботи цього алгоритму для даної задачі в найгіршому випадку?

- ☐ $O(n \log n)$
- ☒ $O(n^2 \log n)$ 
- ☐ $O(n \log n \log n)$
- ☐ $O(n^2)$

ПОЯСНЕННЯ ВІДПОВІДІ


Рекурентне дерево для методу сортування злиттям буде мати висоту $O(\log n)$. Час роботи алгоритму на кожному рівні дерева буде становити $O(n^2)$: кожний рівень потребує n порівнянь і кожне порівняння займає час $O(n)$ в найгіршому випадку. Тож, час роботи алгоритму загалом становитиме: $O(n^2 \log n)$

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 16 (1/1 бал)

Нехай P - це програма, яка реалізує алгоритм швидкого сортування для сортування масиву чисел у зростаючому порядку, при цьому в якості опорного використовується перший елемент масиву. Позначимо через t_1 і t_2 кількість порівнянь елементів в програмі P , які виконуються для вхідних масивів $[1, 2, 3, 4, 5]$ і $[4, 1, 5, 3, 2]$ відповідно. Оберіть правильне твердження з нижче наведених.

- ☐ $t_1 = t_2 = 4$
- ☐ $t_1 = t_2 = 5$
- ☐ $t_1 < t_2$
- ☒ $t_1 > t_2$ 
- ☐ $t_1 = t_2$

ПОЯСНЕННЯ ВІДПОВІДІ

Найгірший випадок для алгоритму швидкого сортування при виборі опорним елементом першого або останнього стається, коли масив вже є відсортованим. Тоді на кожному кроці рекурсії масив розбивається відповідно до співвідношення $T(n) = T(n - 1) + \Theta(n)$.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 17 (1/1 бал)

Який розв'язок має наступне рекурентне співвідношення: $T(n) = T(n/4) + T(n/2) + cn^2$, де $T(1) = c$, $T(0) = 0$ і c - деяка додатня константа?

- ☐ $\Theta(n^3)$
- ☒ $\Theta(n^2)$ ✓
- ☐ $\Theta(n^2 \log n)$
- ☐ $\Theta(n \log n)$

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 18 (1/1 бал)

Яка часова складність наступної рекурсивної функції?

```
DoSomething (ціле число n):  
    if n ≤ 2:  
        return 1  
    else:  
        return DoSomething (floor(sqrt(n)))
```

тут $\text{sqrt}(n) = \sqrt{n}$, а floor - функція, яка повертає цілу частину аргументу.

- ☐ $\Theta(n)$
- ☐ $\Theta(n \log n)$
- ☐ $\Theta(\log n)$
- ☒ $\Theta(\log(\log n))$ ✓

ПОЯСНЕННЯ ВІДПОВІДІ

Рекурентне співвідношення для цієї функції виглядає наступним чином: $T(n) = T(\sqrt{n}) + C$ для $n > 2$. Введемо заміну: $n = 2^m$ і тоді $T(n) = T(2^m) = S(m)$. Тоді рекурентне співвідношення можна переписати як $S(m) = S(m/2) + C$. Його розв'язок за основною теоремою - $S(m) = \Theta(\log m) = \Theta(\log \log n) = T(n)$.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 19 (1/1 бал)

Оберіть правильну асимптотичну часову складність для алгоритму з часом роботи $T(n, n)$, де

$$T(x, c) = \Theta(x) \text{ для } c \leq 2,$$

$$T(c, y) = \Theta(y) \text{ для } c \leq 2,$$

$$T(x, y) = \Theta(x + y) + T(x/2, y/2)$$

☐ $\Theta(n \log n)$

☐ $\Theta(n^2)$

☒ $\Theta(n)$ ✓

☐ $\Theta(n^2 \log n)$

ПОЯСНЕННЯ ВІДПОВІДІ


Рекурентне співвідношення $T(x, y) = \Theta(x + y) + T(x/2, y/2)$ може бути переписане як $T(x, y) = \Theta(x + y) + \Theta(\frac{x+y}{2}) + \Theta(\frac{x+y}{4}) + \Theta(\frac{x+y}{8}) + \dots = \Theta((x + y) + \frac{x+y}{2} + \frac{x+y}{4} + \frac{x+y}{8} + \dots)$. Це буде сума спадної геометричної прогресії із модулем знаменника рівним $1/2$ і вона буде дорівнювати $T(x, y) = \Theta(2(x + y)) = \Theta(x + y)$. А отже $T(n, n) = \Theta(n + n) = \Theta(n)$.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 20 (1/1 бал)

Обхід графу відрізняється від обходу дерева тим, що (оберіть всі можливі варіанти):


- ☒ В графі можуть бути цикли, тому нам потрібно для кожної вершини зберігати чи відвідали ми її вже, чи ні 
- ☐ Обхід вглиб в графі використовує стек, але внутрішній обхід дерева є рекурсивним
- ☐ Обхід вшир в графі використовує чергу, але ефективний обхід вшир в дереві використовує рекурсію

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 21 (1/1 бал)

Граф G має n вершин та m ребер. Якою є точна верхня межа часу роботи алгоритму пошуку вглиб, якщо в ньому використовується матриця суміжностей?

- ☐ $O(n)$
- ☐ $O(n + m)$
- ☒ $O(n^2)$ 
- ☐ $O(nm)$

ПОЯСНЕННЯ ВІДПОВІДІ


Алгоритм пошуку вшир працює за час $O(n + m)$, якщо граф представлений списком суміжностей. У випадку використання матриці суміжностей, яка має розмірність $n \times n$, нам для кожної поточної вершини в алгоритмі потрібно переглянути відповідний їй рядок в матриці - тобто n елементів. Таким чином загальний час роботи становитиме $O(n^2)$.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 22 (1/1 бал)

Якщо до орієнтованого графу G додати додаткове одне ребро, то кількість компонент сильної зв'язності...? (Оберіть найбільш строгі твердження)


- ☐ Не може збільшитись, але може зменшитись
- ☐ Може залишитись тією самою або може змінитись (залежить від графу)
- ☒ Не може зменшитись більше ніж на 1 
- ☐ Не зміниться

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 23 (1/1 бал)

Щоб реалізувати версію алгоритму Дейкстри, яка на незважених графах буде працювати за лінійний час, необхідно використати:

- ☒ Чергу 
- ☐ Стек

- ☐ Піраміду
- ☐ Бінарне дерево пошуку

ПОЯСНЕННЯ ВІДПОВІДІ


Найкоротший шлях у незваженому графі інтерпретується в термінах мінімальної кількості ребер між двома вершинами. Це однаково, якщо приписати всім ребрам вагу 1. Якщо в алгоритмі Дейкстри використовувати звичайну чергу (FIFO), а не чергу з пріоритетами (на основі пірамід), ми зможемо знайти найкоротший шлях за лінійний час $O(n + m)$. Фактично алгоритм Дейкстри в цьому випадку буде працювати аналогічно до алгоритму обходу графу вшир.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 24 (1/1 бал)

Для заданого зваженого графу G та двох вершин s і t було обраховано найкоротший шлях. Серед нижче наведених тверджень оберіть ті, які виконуються:

- ☐ Якщо збільшити вагу всіх ребер у G на 1, то знайдений найкоротший шлях все-одно залишиться тим самим
- ☐ Якщо збільшити вагу всіх ребер у G на 10, то знайдений найкоротший шлях все-одно залишиться тим самим
- ☒ Якщо збільшити вагу всіх ребер у G вдвічі, то знайдений найкоротший шлях все-одно залишиться тим самим 
- ☐ Жодне з переліченого

ПОЯСНЕННЯ ВІДПОВІДІ

Якщо збільшити вагу всіх ребер вдвічі, то найкоротший шлях збережеться, адже це те саме, як перейти з вимірювання довжини в метрах до вимірювання в сантиметрах. Інші варіанти неправильні - див. лекцію 14.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 25 (1/1 бал)

Для заданих двох незростаючих пірамід, кожна з яких має розмірність n , яким є максимальний час побудови незростаючої піраміди, що буде містити всі елементи з обох пірамід?

- ☐ $\Theta(n \log n)$
- ☐ $\Theta(n \log(\log n))$
- ☒ $\Theta(n)$ ✓
- ☐ $\Theta(n^2)$

ПОЯСНЕННЯ ВІДПОВІДІ

Ми можемо побудувати незростаючу піраміду з $2n$ елементів за час $\Theta(n)$ використовуючи наступні кроки. Спочатку створити масив вмісткості $2n$ та скопіювати туди елементи з обох пірамід. Потім запустити процедуру BuildMaxHeap для цього масиву, час роботи якої $\Theta(n)$.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 26 (1/1 бал)

У незростаючій піраміді (Max-Heap), яка містить n елементів, найменший елемент можна знайти за час:

- ☒ $\Theta(n)$ ✓
- ☐ $\Theta(\log n)$
- ☐ $\Theta(\log(\log n))$
- ☐ $\Theta(1)$

Показати відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 27 (1/1 бал)

Тернарна незростаюча піраміда (Max-Heap) аналогічна до бінарної незростаючої піраміди, але в ній кожний внутрішній вузол має не два нащадки, а три. Тернарна піраміда може бути представлена у вигляді масиву A наступним чином. Корінь дерева зберігається в елементі $A[1]$, вузли наступного рівня у порядку зліва направо зберігаються в елементах $A[2]$, ..., $A[4]$. Вузли другого рівня зберігаються у порядку зліва направо починаючи з елемента $A[5]$ і так далі.

Який з наступних масивів представляє правильну тернарну незростаючу піраміду?

- ☐ [1, 3, 5, 6, 8, 9]
- ☐ [9, 6, 3, 1, 8, 5]
- ☐ [9, 3, 6, 8, 5, 1]
- ☒ [9, 5, 6, 8, 3, 1] ✓

ПОЯСНЕННЯ ВІДПОВІДІ

В корені (елемент $A[1]$) зберігається найбільше значення - 9. Наступний рівень містить елементи 5, 6, 8. Елемент 5 має двох нащадків - 3 та 1, які менші за нього. Це задовольняє умові незростаючих пірамід.

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.


ПИТАННЯ 28 (1/1 бал)

Розглянемо схему представлення бінарних дерев за допомогою масиву X . Корінь дерева зберігається в першому елементі масиву $X[1]$. Для внутрішнього вузла дерева, який зберігається в елементі $X[i]$, його лівий нащадок розміщується в елементі $X[2i]$, а правий - в елементі $X[2i + 1]$. Зверніть увагу, що ця схема використовується також для представлення пірамід. Яка повинна бути мінімальна розмірність масиву X для збереження довільного бінарного дерева з n вершин (тут враховані як внутрішні вузли, так і листки).

☐ $\log n$

☐ n

☐ $2^n + 1$

☒ $2^n - 1$ 

Приховати Відповідь


Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 29 (1/1 бал)

Необхідно обрати структуру даних для збереження множини цілих чисел і при цьому обидві нижче наведені операції повинні виконуватись за час $\Theta(\log n)$, де n - кількість елементів в множині:

- Видалення найменшого елементу
- Вставка елементу в множину, якщо його ще немає в множині

Яка з нижче наведених структур даних може використовуватись для цього?

- ☐ Може використовуватись піраміда, але не збалансоване бінарне дерево пошуку
- ☒ Може використовуватись збалансоване бінарне дерево пошуку, але не піраміда 
- ☐ Може використовуватись як піраміда, так і збалансоване бінарне дерево пошуку
- ☐ Ані піраміда, ані збалансоване бінарне дерево пошуку не можуть використовуватись

ПОЯСНЕННЯ ВІДПОВІДІ

Збалансоване бінарне дерево пошуку може знаходити, додавати та видаляти елементи за час $\Theta(\log n)$. Видалення найменшого елементу в такому дереві також буде займати той самий час.

Піраміда - це також збалансоване бінарне дерево, в якому видалення мінімального елементу виконується за час $\Theta(\log n)$, так само як і вставка елементу. Проте для перевірки, чи елемент існує в піраміді потрібно витратити час $\Theta(n)$.

Приховати Відповідь


Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

ПИТАННЯ 30 (1/1 бал)

Хеш-таблиця має 10 комірок та використовує відкриту адресацію для уникнення колізій. Хеш-функція $h(k) = k \bmod 10$ і при цьому застосовується лінійне дослідження. Після вставки шести елементів в початково порожню хеш-таблицю, вона має наступний вигляд:

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

Яка з наступних послідовностей вставки ключів у хеш-таблицю є правильною?

- ☐ 46, 42, 34, 52, 23, 33
- ☐ 34, 42, 23, 52, 33, 46
- ☒ 46, 34, 42, 23, 52, 33 
- ☐ 42, 46, 33, 23, 34, 52

Приховати Відповідь

Ви використали 1 з 1 можливостей надіслати свої матеріали на розгляд.

Показати обговорення



Нове Повідомлення



[Про нас](#) [Преса](#) [FAQ](#) [Контакти](#)

© 2015 Prometheus, some rights reserved

- [Умови надання послуг](#) та [Кодекс Честі](#)

