1. Яка основна характеристика алгоритму, що визначає його ефективність і здатність розв’язувати проблеми?

Кількість деталей

Наявність зайвих кроків

Деталізація

**Передбачуваність результату**

2. Які компоненти структури алгоритму включають в себе вхідні дані та результат обробки?

Тільки вхід

Тільки результат

Вхід і обробка

**Вхід і вихід**

1. Якими способами алгоритми використовуються для покращення ефективності роботи в сфері трансформації даних, згідно з текстом?

Захист інформації від зловмисників

Створення найкоротших маршрутів для GPS-навігації

**Відтворення аудіофайлів у високій якості**

Покращення безпеки на дорогах за допомогою світлофорів

2. Для чого використовуються алгоритми криптографії, згідно з текстом?

Оптимізації роботи світлофорів на перехрестях багатосмугових доріг

Пошуку найефективніших алгоритмів у багатьох сферах застосування

**Захисту конфіденційної інформації та гарантування безпеки в інтернеті**

Генерації випадкових чисел під час проведення лотерей

3. Яка основна мета використання алгоритмів?

Дізнатися всі можливі відповіді, не зважаючи на ситуацію

**Отримати відповідь, яка є балансом між точністю, швидкістю та витратами ресурсів**

Завжди знаходити "ідеальне" рішення в поточній ситуації та контексті

Обробити якнайбільше інформації без урахування потреб або обмежень

4. Який із кроків у створенні алгоритму стосується переведення стратегії в конкретні дії через програмний код?

Визначення проблеми

Розробка стратегії вирішення

**Кодування стратегії**

Тестування алгоритму

5. Яке основне представлення даних використовується в комп'ютерах?

**Десяткова система**

Зображення, текст і відео

Двійкова система

Хеш-таблиці

6. Який із наведених аспектів стосується вивчення підходів, які були вжиті раніше для розв'язання подібних завдань?

Як інші люди створювали нові розв'язки задач

**Якого роду рішення використовувалися для подібних завдань раніше**

Які ресурси для розв'язання задачі є в наявності

Якого роду рішення не приводять до бажаного результату

1. Що таке часова складність алгоритму?

Кількість пам'яті, яку використовує алгоритм під час виконання

Якість алгоритму у розв’язанні задачі обробки вхідних даних обсягу n

**Кількість операцій, які алгоритм виконує під час обробки вхідних даних обсягу n**

Загальна продуктивність алгоритму обробки вхідних даних обсягу n

2. Який із наведених класів часової складності відображає ситуацію, коли алгоритм виконується за сталий час, незалежно від розміру вхідних даних?

O(log n)

O(n)

**O(1)**

1. Яка з наведених алгоритмічних стратегій зберігає результати підзадач для подальшого використання, щоб уникнути повторного розв'язання тих самих підзадач?

Графові алгоритми

Алгоритми сортування

**Динамічне програмування**

Алгоритми пошуку

2. Для якої задачі потрібно вибрати речі таким чином, щоб їх загальна вага не перевищувала вказане обмеження, а загальна вартість була максимальна?

Задача про маршрут

**Задача про рюкзак**

Задача про графи

Задача про множення матриць

3. Який з наступних алгоритмів використовує опорний елемент для впорядкування масиву?

Bubble sort

Insertion sort

**Quicksort**

BFS

1. Що таке структура даних?

**Спосіб зберігання та організації даних**

Система управління даними

Набір алгоритмів для обробки даних

Протокол передачі даних

2. Масиви є найкращими структурами даних

**Для відносно постійних колекцій даних**

Для даних, розмір та структура яких постійно змінюються

Для обох із зазначених вище ситуацій

Для жодної із зазначених вище ситуацій

1. Над стеком виконали таку послідовність операцій. Push(1), Push(2), Pop, Push(1), Push(3), Pop, Pop, Pop, Що зараз знаходиться на вершині стеку?

1

2

3

**Стек порожній**

2. Що означає принцип LIFO?

Last In Furthest Out

Least In First Out

**Last In First Out**

Least In Furthest Out

3. У якому порядку видаляються елементи зі стеку?

**Зворотному**

Ієрархічному

Альтернативному

Послідовному

4. Як прийнято називати операцію вставки у стек?

Insert

**Push**

Рop

Put

За допомогою якого методу виконується операція видалення з черги в бібліотеці queue?

**get**

put

pop

list

2. Якщо елементи «A», «B», «C» і «D» помістити в чергу та видалити по одному, у якому порядку вони будуть видалені?

**ABCD**

DCBA

DCAB

ACBD

Результат

Які з наведених нижче структур даних дозволяють вставляти та видаляти елементи з обох сторін? (можливо вибрати кілька правильних відповідей)

Стек (Stack)

**Двостороння черга (deque)**

Черга (queue)

**Масив (array)**

1. Зв'язаний список — це така структура даних, де?

**Кожен елемент містить дані та посилання на наступний елемент**

Всі елементи зберігаються у відсортованому порядку

Доступ до елементів відбувається за індексом

Кожен елемент має два посилання: на наступний та попередній елементи

2. На вашу думку, яка основна перевага зв'язаного списку в порівнянні з масивом?

Швидкий доступ до елементів за індексом

Можливість автоматичного сортування елементів

Можливість зберігати різні типи даних

**Відсутність потреби в перерахуванні місця в пам'яті при додаванні/видаленні елементів**

3. Яка складність додавання нового елемента в початок зв'язаного списку?

**O(1)**

O(n)

O(log n)

4. Як визначається кінець зв'язаного списку?

**Посиланням на None**

Посиланням на останній елемент

Індексацією

Посиланням на перший елемент

Окрім компонентів інформації та посилання, які інші компоненти також має містити двозв’язний список?

Інформацію сортування про список

**Покажчики голови та хвоста на перший та останній вузли**

Поточний вузол, до якого востаннє зверталися

Перша та третя відповіді

1. Що таке двозв'язний список?

Список, де кожен елемент має посилання тільки на наступний елемент

**Список, де кожен елемент має посилання на наступний та попередній елементи**

Список, де всі елементи зберігаються у відсортованому порядку

Список, де доступ до елементів відбувається за індексом

2. Яка основна перевага двозв'язного списку, порівняно зі зв'язаним списком?

Двозв'язний список дозволяє вставляти та видаляти вузли швидше

Двозв'язний список займає менше пам'яті

**Двозв'язний список дозволяє обходити список в обидва боки**

Двозв'язний список дозволяє швидший доступ до елементів за індексом

3. Що представляє вузол у двозв'язному списку?

Тільки дані без посилань

Дані та посилання тільки на наступний вузол

Посилання тільки на попередній вузол

**Дані, посилання на наступний вузол та посилання на попередній вузол**

1. Яка структура даних представляє набір пар ключ-значення?

Черга

Стек

Дерево

**Хеш-таблиця**

2. Яка структура даних використовується для представлення мережі взаємопов’язаних елементів?

Черга

Стек

Зв’язаний список

**Граф**

3. Яка структура даних використовується для ефективних операцій пошуку, вставки та видалення в постійному часі?

Черга

Стек

Зв’язаний список

**Хеш-таблиця**

4. Вузлам (або вершинам) графа відповідають:

Відношення між об'єктами

**Об'єкти**

Зв'язки

Типи відношень

Результат

5. Ребрам графа відповідають:

Канали

Типи відношень

Об'єкти

**Відношення між об'єктами**

6. Яка структура даних представляє ієрархічну структуру з набором пов’язаних вузлів?

Черга

Стек

**Дерево**

Граф

7. Як називається елемент дерева, який не посилається на інші?

Корінь

**Лист**

Вузол

Проміжний

Що таке рекурсія в програмуванні?

Це процес повторення коду в циклі

**Це метод розв’язання проблеми, коли функція викликає саму себе**

Це спосіб зберігання даних у пам'яті

Це структура даних для представлення ієрархічних відношень

2. Які можливі негаразди може спричинити неправильно сконструйована рекурсивна функція?

Нескінченна рекурсія

Переповнення стеку

**Обидва варіанти вище**

Немає правильної відповіді

3. Що таке базовий випадок у рекурсії?

**Це умова, яка завершує рекурсивні виклики**

Це значення, яке повертається функцією на кожному кроці

Це кількість рекурсивних викликів, які виконує функція

Це умова, яка ініціює перший рекурсивний виклик

4. Чому мемоізація корисна в рекурсивних алгоритмах?

Вона зменшує кількість рекурсивних викликів

**Вона зберігає проміжні результати для уникнення повторних викликів**

Вона автоматично оптимізує рекурсивну функцію

Вона зменшує складність алгоритму в загальному випадку

5. Чому рекурсія може бути менш ефективною, ніж ітерація?

**Тому що вона вимагає більше пам'яті для зберігання стеку викликів**

Тому що вона виконується повільніше

Тому що рекурсія використовує більше коду

Тому що вона не може розв’язувати проблеми, які можна розв’язати ітерацією

1. Як ви думаєте, як швидше відсортувати масив, використовуючи бульбашковий метод, розглянутий у конспекті?

За зростанням елементів

За спаданням елементів

**Однаково**

2. Масив сортується бульбашковим методом за зростанням: найбільший елемент має опинитися в кінці (вгорі) масиву. За скільки проходів по масиву найбільший елемент у масиві опиниться вгорі? Якщо n — це число елементів масиву

**За 1 прохід**

За n-1 проходів

За n проходів

3. Яка складність алгоритму бульбашкового сортування?

Лінійна

Логарифмічно-лінійна

**Квадратична**

Логарифмічна

1. Прикладом якого сортування є сортування вставкою?

**Розкладання колоди карток, вибираючи одну за одною**

Розкидання монет по кишенях

Розміщення книг на полиці в алфавітному порядку

Розташування коробок на складі за розміром

1. Що ж, запам’ятали, яка складність алгоритму сортування прямою вставкою?

Лінійна

Логарифмічно-лінійна

**Квадратична**

Логарифмічна  
  
1. Яка техніка використовується у швидкому сортуванні?

**“Розділяй та володарюй”**

Бінарний пошук

Backtrack

Принцип винесення рішень

2. Яка часова складність алгоритму швидкого сортування у найгіршому випадку?

Лінійна

Логарифмічно-лінійна

**Квадратична**

Логарифмічна

1. Яка основна ідея сортування злиттям (Merge Sort)?

**Розбити масив на дві половини, відсортувати кожну з них окремо, а потім злити їх в один відсортований масив.**

Пройтися по масиву та поміняти місцями кожні два сусідні елементи, якщо вони в неправильному порядку.

Відсортувати першу половину масиву, а потім використати цю інформацію для сортування другої половини.

Знайти мінімальний елемент у масиві та перемістити його на правильне місце.

2. Яка часова складність алгоритму сортування злиттям?

Лінійна

**Логарифмічно-лінійна**

Квадратична

Логарифмічна

Результат

1. За методом сортування Шелла, які елементи масиву порівнюються?

Сусідні

Крайні

Протилежні

**Віддалені один від одного з певним кроком**1. Який принцип роботи лежить в основі алгоритму порозрядного сортування?

**Сортування кожного розряду від меншого до більшого**

Сортування кожного розряду від більшого до меншого

Обмін місцями сусідніх елементів

Рекурсивне розбиття та злиття масиву

3. Що з наведеного не є прикладом алгоритму «розділяй і володарюй»?

Сортування злиттям

Швидке сортування

Двійковий пошук

**Бульбашкове сортування**

Результат

4. Які елементи порівнюють у методі сортування бульбашки?

**Сусідні**

Віддалені один від одного з певним кроком

Крайні

Протилежні

5. Який засіб програмування використовується для реалізації методу швидкого сортування?

Сканування масиву

Пошук максимального значення

**Рекурсія**

1. Який з наступних алгоритмів є прикладом лінійного пошуку?

Двійковий пошук

Пошук за допомогою хеш-таблиці

**Пошук першого входження елемента в масиві**

Пошук у бінарному дереві

2. Яка найгірша часова складність алгоритму лінійного пошуку?

Константна

Логарифмічна

**Лінійна**

Квадратична

3. Яка найкраща часова складність алгоритму бінарного пошуку?

Константна

**Логарифмічна**

Лінійна

Квадратична

4. У чому суть бінарного пошуку?

**Знаходження елемента шляхом ділення масиву навпіл кожен раз, поки елемент не буде знайдений або поки масив не стане порожнім.**

Знаходження елемента шляхом послідовного обходу кожного елемента масиву від початку до кінця.

Знаходження елемента в масиві за допомогою хешування.

Знаходження елемента в масиві за допомогою випадкового вибору позицій.

1. Яка з наведених операцій НЕ є характерною для хеш-таблиці?

Пошук

Вставка

Видалення

**Відсортування**

2. Що перетворює ключі в хеш-таблиці на індекси?

Метод ланцюжків

Лінійне зондування

**Хеш-функція**

Колізія

3. Яке з наведених нижче є характерною особливістю хеш-функції?

Можливість зміни

**Детермінованість**

Кластеризація

Колізія

4. Як називається проблема в хеш-таблицях, коли два різні ключі продукують одне й те саме значення хешу?

Лінійне зондування

Хешування

**Колізія**

Ланцюгова адресація

5. Який з наведених методів використовується для розв’язання колізій у хеш-таблицях, коли ключі об'єднуються в один "ланцюг"?

Лінійне зондування

Хешування

**Метод ланцюжків**

Двійкове дерево  
  
1. У якому порядку підрядок порівнюють з текстом в алгоритмі Боєра-Мура?

**З кінця підрядка**

З початку підрядка

З середини підрядка

Початок випадковий  
  
1. Який алгоритм пошуку в рядках працює шляхом перевірки кожного можливого місця в тексті, де підрядок може починатися?

**Наївний алгоритм**

Алгоритм Кнута-Морріса-Пратта

Алгоритм Боєра-Мура

Алгоритм Рабіна-Карпа

2. Який алгоритм використовує префіксну функцію для ефективного пошуку входжень підрядка в рядок?

Наївний алгоритм

**Алгоритм Кнута-Морріса-Пратта**

Алгоритм Боєра-Мура

Алгоритм Рабіна-Карпа

3. Який із алгоритмів використовує таблицю зсувів і спробує просунути якомога більше символів відповідно до останнього символу підрядка, якщо не відбулося збігу?

Наївний алгоритм

Алгоритм Кнута-Морріса-Пратта

**Алгоритм Боєра-Мура**

Алгоритм Рабіна-Карпа

4. Який алгоритм використовує хешування для порівняння підрядків замість порівняння кожного символу окремо?

Наївний алгоритм

Алгоритм Кнута-Морріса-Пратта

Алгоритм Боєра-Мура

**Алгоритм Рабіна-Карпа**

5. Який алгоритм при порівнянні підрядків зазвичай починає порівняння з кінця підрядка?

Наївний алгоритм

Алгоритм Кнута-Морріса-Пратта

**Алгоритм Боєра-Мура**

Алгоритм Рабіна-Карпа

6. У якому алгоритмі частина підрядка, що не збіглася, не перевіряється повторно при подальшому зсуві підрядка вздовж тексту?

Наївний алгоритм

**Алгоритм Кнута-Морріса-Пратта**

Алгоритм Боєра-Мура

Алгоритм Рабіна-Карпа  
  
1. Яке визначення графа в математиці?

Послідовність чисел або символів

**Множина вершин і ребер, які їх з'єднують**

Геометрична фігура, схожа на дерево

Абстрактний тип даних

2. Що являє собою ступінь вершини у графі?

Загальна кількість вершин у графі

Загальна кількість ребер у графі

**Кількість ребер, які виходять з даної вершини**

Сумарна вага ребер у графі

3. Що таке орієнтований граф?

Граф, у якому ребра не мають напрямку

**Граф, у якому ребра мають напрямок**

Граф, у якому всі вершини з'єднані між собою

Граф, у якому тільки вершини без ребер

4. Яка основна характеристика матриці суміжності при представленні графа?

Кожен елемент матриці відображає відстань між двома вершинами

**Кожен елемент матриці відображає наявність або відсутність ребра між двома вершинами**

Кожен елемент матриці відображає загальну кількість вершин у графі

Кожен елемент матриці відображає загальну кількість ребер у графі

5. Що являє собою список суміжності при представленні графа?

Список усіх вершин графа

Список усіх ребер графа

**Список вершин, з якими пов'язана кожна вершина**

Список ваг ребер графа

6. Як представлено ребро у списку ребер?

Одним числом, що вказує на вагу ребра

Іменем ребра

Індексом ребра у списку

**Парою вершин**

7. Яку структуру даних зазвичай використовують для ітеративної реалізації алгоритму пошуку в глибину (DFS)?

Масив

Зв’язаний список

**Стек**

Черга

8. Яка структура даних зазвичай використовується для ітеративної реалізації алгоритму пошуку в ширину (BFS)?

Масив

Зв’язаний список

Стек

**Черга**

9. Що з наведеного нижче не є структурою даних, яка зазвичай використовується для реалізації графа?

Матриця суміжності

Список суміжності

**Хеш-таблиця**

Список ребер

1. Якщо G — орієнтовний граф з 20 вершинами, скільки булевих значень потрібно для представлення G за допомогою матриці суміжності?

**400**

40

200

20

2. Для якого з наступних випадків краще використовувати алгоритм BFS замість DFS?

**Для знаходження найкоротшого шляху в незваженому графі**

Для виявлення циклів у незваженому графі

Для топологічного сортування графа

Для визначення зв'язності незваженого графа

3. У якому випадку алгоритм Дейкстри буде працювати некоректно?

Граф має додатні ваги ребер

Граф є неорієнтованим

Граф є повністю зв'язним

**Граф містить від'ємні ваги ребер**

4. Як створити порожній неорієнтований граф за допомогою бібліотеки networkx?

**nx.Graph()**

nx.DiGraph()

nx.create\_graph()

nx.empty\_graph()

5. Як додати ребро між вершинами 'A' та 'B' з вагою 3 в граф G за допомогою бібліотеки networkx?

G.edge('A', 'B', weight=3)

**G.add\_edge('A', 'B', weight=3)**

G.add('A', 'B', 3)

G.create\_edge('A', 'B', 3)

6. Яка основна ідея алгоритму PageRank?

Розрахунок кількісних характеристик вебсторінок

**Визначення важливості вебсторінок на основі взаємного посилання**

Фільтрація вмісту вебсторінок за ключовими словами

Індексація вебсторінок для пошукових систем

Що таке вузол дерева в інформатиці?

З'єднання між двома елементами

Кінцевий елемент дерева

**Елемент із дочірніми та батьківським вузлами**

Процес обходу дерева

Який останній крок у зворотному (postorder) обході дерева?

Обхід лівого піддерева

Обхід правого піддерева

**Відвідування кореневого вузла**

Відвідування батьківського вузла

Що вдається ефективно представляти за допомогою дерев в інформатиці?

Лінійні відношення

**Ієрархічну інформацію**

Хешовані таблиці

Дані для сортування

Чому дерева повільніші за хеш-таблиці?

они потребують більше пам'яті

У них відсутня індексація

**Усі операції дерев з лінійним часом виконання**

Через відсутність впорядкованості даних

За якої умови дерево називається бінарним (двійковим)?

Кожен вузол має рівно двох нащадків

**Кожен вузол має не більше двох нащадків**

Усі вузли дерева мають нащадків

Глибина всіх вузлів однакова

Що є ключем у вузлі двійкового дерева пошуку?

**Значення, яке ідентифікує вузол**

Посилання на інший вузол

Правило для пошуку в дереві

Тип вузла в дереві

Яка операція у двійковому дереві пошуку є найскладнішою?

Вставка

Пошук

**Видалення**

Оновлення

Що відбувається у найгіршому випадку двійкового дерева пошуку при вставці елементів у впорядкованому порядку?

Дерево стає збалансованим

**Дерево деградує в посортований список**

Час виконання операцій скорочується

Усі вузли стають листовими вузлами

1. Яка основна особливість AVL-дерева?

**Різниця у висоті піддерев кожного вузла не перевищує 1**

Різниця у висоті піддерев кожного вузла не перевищує 2

Кожен вузол має рівно двох нащадків

Глибина всіх вузлів однакова

2. Які операції можуть викликати перебалансування AVL-дерева?

Вставка, видалення та пошук

Пошук та видалення

**Вставка та видалення**

Тільки вставка

3. Чому AVL-дерева важливі в контексті структур даних?

**Вони зменшують час виконання операцій пошуку, вставки та видалення**

Вони автоматично сортують дані

Вони гарантують безпечне зберігання даних

Вони зменшують використання пам'яті

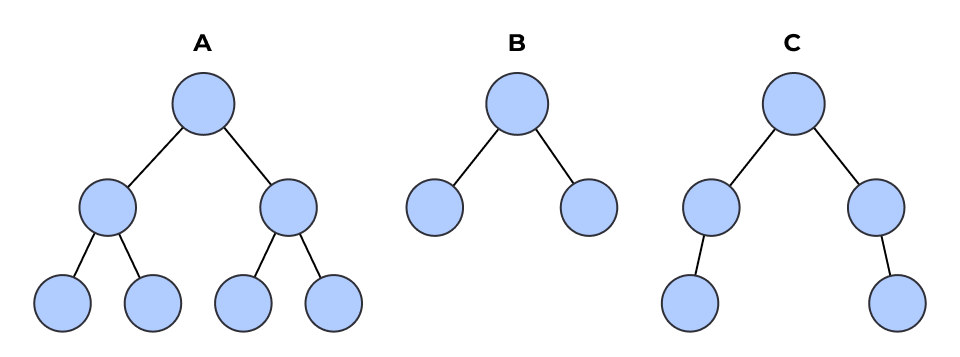
4. Яка є кількість видів повороту для перетворення двійкового дерева пошуку на збалансоване?

2

3

**4**

5



1. Яке з дерев є бінарним?

A

B

C

**Усі дерева**2. Який колір має корінь у червоно-чорному дереві?

Червоний

**Чорний**

Білий

Сірий

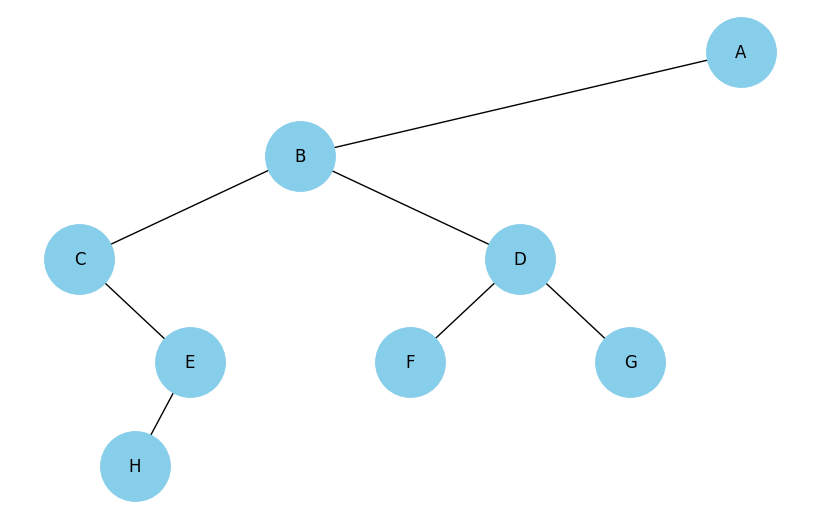
3. Яка основна відмінність 2-3-дерев від інших бінарних дерев?

**Вони мають більше гнучкості в балансуванні**

Вони завжди мають фіксовану висоту

Вони використовують кольори для балансування

Вони мають високий порядок гілкування



5. Дайте відповідь на питання стосовно зображеного дерева. Скільки листків є в дереві?

**3**

4

5

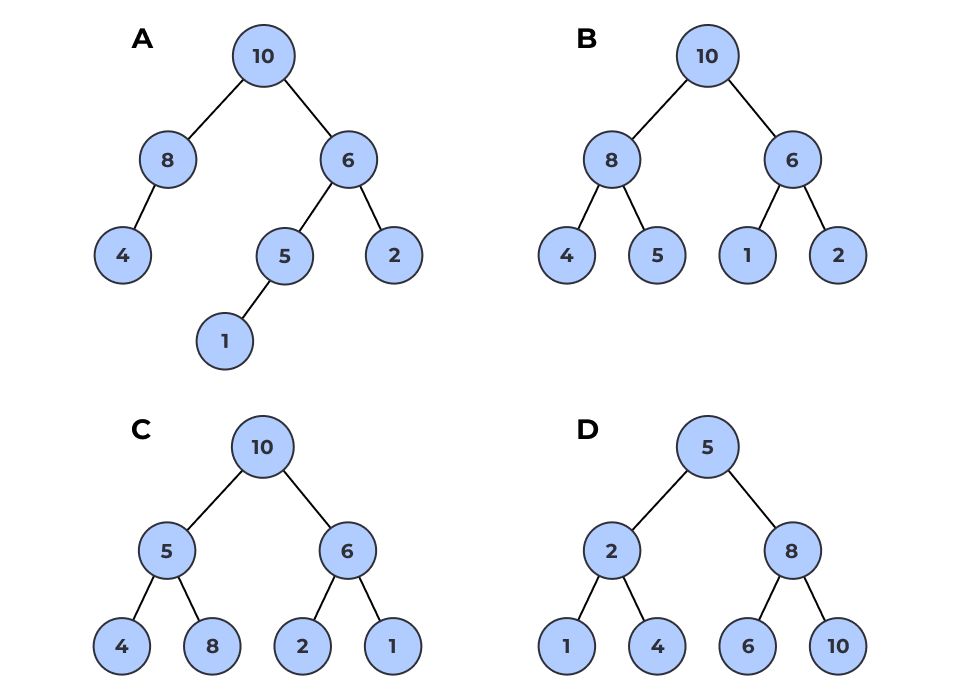
8

6. Дайте відповідь на питання стосовно зображеного дерева. Які елементи є нащадками вузла “B”?  
C, D

**C, E, H, D, F, G**

E, F, G

E, H, F, G



1. Максимальна купа — це купа, у якій значення кожного батьківського елемента більше або дорівнює значенням його дочірніх елементів. Що з наведеного вище є максимальною купою?

А

**B**

C

D

2. Яка структура даних використовується для ефективного пошуку мінімальних і максимальних елементів?

Стек

Черга

Зв’язаний список

**Купа**

3. Яка часова складність операції побудови купи (Build Heap)?

Лінійна логарифмічна

Квадратична

Логарифмічна

**Лінійна**

4. Які умови необхідні для того, щоб дерево було купою?

Дерево має бути повним двійковим деревом

Для кожного вузла дерева значення повинно бути більше або менше за значення його дочірніх вузлів

**Мають виконуватися обидві вищезазначені умови — 1 та 2**

Жодна з вище наведених умов не є обов'язковою

5. Яка характеристика описує максимальну купу?

Корінь містить найменший елемент

Кожен вузол має бути меншим за його дітей

**Корінь містить найбільший елемент**

Усі елементи на нижньому рівні мають бути більші за їхні батьківські вузли

6. Що визначається як "останній вузол" у контексті купи?

Кореневий вузол

Найбільший вузол

Крайній лівий елемент нижнього рівня

**Крайній правий елемент нижнього рівня**

7. Яка часова складність операції вставки в купу?

Константна

**Логарифмічна**

Лінійна

Лінійно-логарифмічна

8. Що відбувається під час операції видалення в купі?

Видаляється випадковий елемент і замінюється останнім вузлом

**Видаляється кореневий елемент і замінюється останнім вузлом**

Видаляється останній елемент і замінюється найменшим вузлом

Видаляється найменший елемент і замінюється останнім вузлом

10. Який метод у модулі heapq використовується для додавання елемента до купи?

heapify()

**heappush()**

heappop()

heapreplace()

.

11. Що робить функція heapify() в модулі heapq?

Додає новий елемент до купи

Видаляє максимальний елемент із купи

**Перетворює список у бінарну купу**

Знаходить найбільший елемент у купі

Який з алгоритмів НЕ відноситься до жадібних?

Алгоритм мінімального остовного дерева Прима

Алгоритм мінімального остовного дерева Краскала

**Алгоритм найкоротшого шляху всіх пар Флойда-Воршала**

Алгоритм мінімального остовного дерева Дейкстри

Який із цих алгоритмічних підходів намагається досягти локалізованого оптимального рішення?

**Жадібний підхід**

Підхід «розділяй і володарюй»

Динамічний підхід

Усе перераховане вище

Різновидом чого є алгоритм Крускала для знаходження мінімального остовного дерева?

Динамічного програмування

**Жадібного алгоритму**

Спеціальної проблеми

Жодного з вказаних

Що з наведеного не є прикладом жадібного алгоритму?

Алгоритм Дейкстри

Алгоритм Краскала

Алгоритм Прима

**Пошук у глибину**

Який алгоритм розв’язує задачу найкоротшого шляху для всіх пар?

Алгоритм Дейкстри

**Алгоритм Флойда-Воршала**

Алгоритм Прима

Алгоритм Краскала  
  
1. Яка перевага лінійного програмування?

Здатність знаходження найкращого рішення для всіх можливих задач.

Можливість точного опису всіх реальних задач.

Здатність обробляти нелінійні цільові функції та обмеження.

**Здатність знаходження найкращого рішення для задач, що можуть бути точно описані лінійними моделями.**

2. Що являє собою симплекс-метод у лінійному програмуванні?

Метод для генерації випадкових чисел.

Алгоритм для розв'язання систем лінійних рівнянь.

**Метод для знаходження розв'язку лінійної оптимізаційної задачі.**

3. Які типи змінних можна визначити в PuLP?

Неперервні

Цілочисельні

**Неперервні і цілочисельні**

Дійсні

4. Що визначають обмеження типу "кордони" в задачі лінійного програмування?

Задають умови, які повинні бути дотримані цільовою функцією.

**Це обмеження, які явно накладаються на вхідні значення.**

Обмеження, що включають більше однієї змінної.

Визначають цільову функцію задачі.

5. Як можна перевірити статус розв'язку задачі в PuLP після виклику model.solve()?

pulp.checkStatus(model)

model.solutionStatus()

**pulp.LpStatus[model.status]**

model.checkSolveStatus()

6. Що означає статус "Infeasible" при розв'язанні задачі лінійного програмування?

Задача успішно розв'язана з оптимальним рішенням.

**Не існує жодного набору змінних, який би задовольняв усі обмеження.**

Задача має необмежений розв'язок.

Статус розв'язку не визначено.

7. Що таке рандомізовані алгоритми?

Алгоритми, які використовують випадкові числа для ускладнення роботи.

**Алгоритми, які використовують випадковість як частину свого виконання.**

Алгоритми, які генерують випадкові вхідні дані для прийняття рішень.

Алгоритми, які випадковим чином обирають свою складність.

8. Яка основна ідея тесту Рабіна-Міллера для перевірки простоти числа?

Використання випадкових чисел для генерації чисел Фібоначчі.

Перевірка числа на парність і ділення на 3.

**Використання модульної арифметики та малої теореми Ферма.**

Визначення простоти числа за допомогою тесту простоти Ферма.

10. Яка основна ідея методу Монте-Карло?

**Апроксимація результатів за допомогою випадкових зразків.**

Точне розв'язання складних задач.

Використання теорії ймовірностей для всіх обчислень.

Застосування алгоритмів глибинного навчання.

11. Яке обмеження методу Монте-Карло?

Недостатнє використання квантових обчислень.

**Залежність від кількості випадкових даних та експериментів.**

Відсутність можливості застосування в математиці.

Точність результатів залежить від можливостей мови програмування.