Вопрос 1

Ответ сохранен

В 2-3 дереве узел может иметь...

- [ ] a. ... только 2 ключа.
- [ ] b. ... ровно 2 ключа и 3 потомка.
- [x] c. ... только 1 или 2 ключа и, соответственно, 2 или 3 потомка.
- [ ] d. ... от 2 до 3 ключей.

Вопрос 2

Отметить вопрос

Ответ сохранен

В AVL-дереве после вставки нового узла...

- [ ] a. ... выполняется балансировка с помощью перекрашивания узлов.
- [x] b. ... выполняется балансировка с помощью поворотов, если коэффициент сбалансированности стал равен -2 или 2.
- [ ] c. ... дерево перестраивается в полное бинарное дерево.
- [ ] d. ... дерево всегда остается идеально сбалансированным.

Вопрос 3

Ответ сохранен

При одинарном левом повороте в AVL-дереве для исправления ситуации "left-left"...

- [x] a. ... левый потомок проблемного узла становится новым корнем поддерева.
- [x] b. ... баланс восстанавливается за одну операцию.
- [ ] c. ... правое поддерево становится выше.
- [ ] d. ... требуется изменить цвет узлов.

Вопрос 4

Отметить вопрос

Ответ сохранен

Основное преимущество B+-дерева перед B-деревом для использования в качестве индекса в базах данных заключается в том, что...

- [ ] a. ... B+-дерево требует меньше памяти.
- [x] b. ... в B+-дереве все данные хранятся только в листьях, что ускоряет последовательный обход.
- [ ] c. ... B+-дерево проще реализовать.
- [ ] d. ... B+-дерево имеет меньшую высоту.

Вопрос 5

Отметить вопрос

Ответ сохранен

Что хранится в листьях B+-дерева, в отличие от B-дерева?

- ( ) a. Указатели на корни других деревьев.
- (•) b. Все данные (значения), связанные с ключами, образуя плотный последовательный индекс.
- ( ) c. Только ключи без associated данных.
- ( ) d. Служебная информация для балансировки.

Очистить мой выбор

Вопрос 6

Отметить вопрос

Ответ сохранен

При вставке нового ключа в B-дерево...

- [x] a. ... если узел переполняется, он разделяется (split).
- [x] b. ... вставка всегда происходит в листовой узел.
- [ ] c. ... используется перекрашивание узлов для балансировки.
- [x] d. ... дерево растет в высоту только когда разделяется корень.

Вопрос 7

Отметить вопрос

Ответ сохранен

Чем B-дерево принципиально отличается от сбалансированных бинарных деревьев поиска (например, AVL)?

- [x] a. B-дерево оптимизировано для систем, работающих с диском (блочная структура).
- [x] b. Высота B-дерева обычно меньше, чем у бинарного дерева с тем же количеством ключей.
- [x] c. B-дерево может иметь более двух потомков у одного узла.
- [ ] d. B-дерево не является деревом поиска.

Вопрос 8

Отметить вопрос

Ответ сохранен

Какое свойство является самым важным для бинарного дерева поиска (BST)?

- [x] a. Для любого узла все значения в левом поддереве меньше, а в правом — больше.
- [ ] b. Это полное бинарное дерево.
- [ ] c. Каждый узел имеет не более двух потомков.
- [ ] d. Высота дерева всегда сбалансирована.

Вопрос 9

Отметить вопрос

Ответ сохранен

Для какого типа обхода бинарного дерева последовательность посещения узлов будет соответствовать возрастанию ключей в бинарном дереве поиска (BST)?

- ( ) a. Обратный (post-order) обход
- (•) b. Симметричный (in-order) обход
- ( ) c. Прямой (pre-order) обход
- ( ) d. Обход в ширину (level-order)

Очистить мой выбор

Вопрос 10

Отметить вопрос

Ответ сохранен

Для чего обычно используется структура данных "дек" (двусторонняя очередь)?

- ( ) a. Для реализации стека.
- ( ) b. Для реализации очереди с приоритетом.
- ( ) c. Для реализации алгоритма обхода графа в ширину (BFS).
- (•) d. В ситуациях, когда добавление и удаление элементов требуется с обоих концов.

Вопрос 11

Отметить вопрос

Ответ сохранен

Какие из следующих операций для дека, реализованного на двусвязном списке, выполняются за O(1)?

- [ ] a. `peekMiddle` (доступ к среднему элементу)
- [ ] b. `find` (поиск элемента по значению)
- [x] c. `pushFront` (добавление в начало)
- [x] d. `popBack` (удаление с конца)

Вопрос 12

Отметить вопрос

Ответ сохранен

Какие из следующих свойств обязательны для красно-черного дерева (RBT)?

- [ ] a. Высота поддеревьев любого узла различается не более чем на 1.
- [x] b. Если узел красный, то оба его потомка черные.
- [x] c. Все листья (NIL) являются черными.
- [x] d. Корень дерева всегда черный.
- [x] e. Каждый узел либо красный, либо черный.

Вопрос 13

Отметить вопрос

Ответ сохранен

В красно-черном дереве свойство "все пути от любого узла до его листьев содержат одинаковое количество черных узлов" гарантирует, что...

- [ ] a. ... корень дерева всегда красный.
- [x] b. ... самый длинный путь от корня к листу не более чем в два раза длиннее самого короткого.
- [ ] c. ... операции вставки и удаления всегда выполняются за O(1).
- [ ] d. ... дерево является идеально сбалансированным.

Вопрос 14

Отметить вопрос

Ответ сохранен

Какие из следующих структур данных считаются линейными?

- [ ] a. Хеш-таблица
- [x] b. Очередь
- [ ] c. Бинарное дерево поиска
- [x] d. Стек

Вопрос 15

Отметить вопрос

Ответ сохранен

Какая из структур данных НЕ подходит для реализации очереди с приоритетом, где нужно быстро удалять элемент с наивысшим приоритетом?

- ( ) a. Сбалансированное бинарное дерево поиска
- (•) b. Несортированный массив
- ( ) c. Двоичная куча (min-heap или max-heap)
- ( ) d. Отсортированный массив

Очистить мой выбор

Вопрос 16

Отметить вопрос

Ответ сохранен

Какие из утверждений об очереди верны?

- [x] a. Очередь с приоритетом может быть реализована на основе кучи.
- [ ] b. В очереди удаление элемента возможно с любого конца.
- [x] c. Циклическая очередь позволяет эффективно использовать память при реализации на массиве.
- [x] d. Очередь реализует принцип FIFO (First-In, First-Out).

Вопрос 17

Отметить вопрос

Ответ сохранен

Какие из этих операций могут быть выполнены за время O(1) в очереди, реализованной на основе связного списка?

- [x] a. Вставка элемента в конец (enqueue).
- [ ] b. Доступ к произвольному элементу по индексу.
- [x] c. Удаление элемента из начала (dequeue).
- [ ] d. Поиск элемента по значению.

Вопрос 18

Отметить вопрос

Ответ сохранен

Какие из этих деревьев являются самобалансирующимися?

- [x] a. AVL-дерево
- [ ] b. Простое бинарное дерево поиска
- [x] c. 2-3 дерево
- [x] d. Красно-черное дерево

Вопрос 19

Отметить вопрос

Ответ сохранен

В каких случаях использование стека является наиболее подходящим?

- [x] a. Обход графа в глубину (DFS).
- [x] b. Реализация отмены (Undo) операции в текстовом редакторе.
- [ ] c. Реализация кеша, где нужно вытеснять самые старые элементы.
- [x] d. Вычисление арифметических выражений в обратной польской записи.

Вопрос 20

Отметить вопрос

Ответ сохранен

Какие из следующих утверждений о стеке являются верными?

- [x] a. Стек реализует принцип LIFO (Last-In, First-Out).
- [x] b. Основные операции со стеком — `push`, `pop` и `peek`.
- [x] c. Стек может быть эффективно реализован на основе односвязного списка.
- [x] d. Рекурсия использует стек вызовов для хранения состояний функции.