

Вопросы теоретического анализа

В разделе приведены вопросы, используемые на полном экзамене. При распределении вопросов студент получает номера вопросов из данных списков.

Блок "Удовлетворительно"

1. Дайте формальное математическое определение O - и Ω -нотаций.
2. Дайте формальное математическое определение O - и Θ -нотаций.
3. Дайте формальное математическое определение Θ - и Ω -нотаций.
4. Объясните разницу между худшим случаем и лучшим случаем (O - и Ω -нотация) на примере алгоритма быстрой сортировки.
5. Объясните разницу между средним случаем и лучшим случаем (Θ - и Ω -нотация) на примере алгоритма сортировки вставками.
6. В чем разница между абстрактным типом данных и структурой данных? Приведите пример для очереди и деки.
7. Перечислите преимущества и недостатки односвязного списка по сравнению со статическим массивом.
8. Перечислите преимущества и недостатки двунаправленного списка по сравнению с динамическим массивом.
9. Обоснуйте, почему вставка в середину связного списка формально занимает $O(1)$, а на практике работает гораздо медленнее?
10. Опишите алгоритм сортировки подсчетом. В каких случаях она эффективна?
11. Почему сортировка пузырьком считается неэффективной? Назовите ее сложность.
12. Опишите алгоритм BFS (поиск в ширину). Какую структуру данных он использует и почему?
13. Опишите алгоритм DFS (поиск в глубину). Какую структуру данных он использует и почему?
14. Опишите алгоритм Дейкстры с точки зрения жадных алгоритмов.
15. Опишите алгоритм Дейкстры с использованием матрицы смежности.
16. Что такое хэш-функция и коллизия?
17. Что такое коллизия в хеш-таблице и какие есть способы борьбы с ней?
18. Что такое коэффициент заполнения, где и как он применяется? Для чего нужно перехэширование?
19. Перечислите преимущества и недостатки сбалансированного дерева поиска по сравнению с несбалансированным?
20. Что такое высота дерева? Докажите оценку высоты AVL и красно-черного дерева из n элементов.

Блок "Хорошо / Отлично"

1. Приведите пример алгоритма, где O , Θ и Ω -оценки различаются. Объясните, почему на практике важнее знать O -оценку, хотя она «пессимистична».
2. Объясните идею метода потенциалов для амортизационного анализа на примере динамического массива (при переполнении массив удваивается).
3. Сравните методы бухгалтерского учета и агрегирования. Какой из них понятнее для

программиста и почему?

4. Проанализируйте, как выбор реализации очереди на массиве (кольцевой буфер) против реализации на связном списке влияет на кэш-эффективность и сложность операций.
5. Почему в развернутом связном списке размер блока (k) выбирают фиксированным (16-64), а не вычисляют как $\sqrt{n}$?
6. Для сценария «частые вставки/удаления в начале и последовательный обход» выберите оптимальную структуру из: массив, односвязный список, двусвязный список, развернутый список. Обоснуйте.
7. Для сценария «частые случайные доступы по индексу, редкие модификации» выберите оптимальную структуру из: массив, односвязный список, двусвязный список, развернутый список. Обоснуйте.
8. Для сценария «непредсказуемый паттерн доступа с преобладанием операций в середине структуры» выберите оптимальную структуру из: массив, односвязный список, двусвязный список, развернутый список. Обоснуйте.
9. Опишите идею XOR-связного списка, почему он не получил широкого распространения, несмотря на экономию памяти?
10. Объясните влияние выбора последовательности шагов (последовательность Шелла, последовательность Седжвика) на сложность сортировки Шелла.
11. Докажите, что TimSort работает за $O(n \log n)$ в худшем случае. Для чего нужен режим «галоп»?
12. Докажите корректность LSD-сортировки методом математической индукции.
13. Почему Radix sort может работать быстрее, чем $O(n \log n)$, и не противоречит ли это теореме о нижней оценке для сортировок сравнениями?
14. Проведите амортизационный анализ методом бухгалтерского учета операции вставки и удаление в хеш-таблицу с цепочками при динамическом перехэшировании.
15. Докажите, что BFS гарантированно находит кратчайший путь в невзвешенном графе.
16. Сравните временную сложность BFS для графа, представленного матрицей смежности и списками смежности. Почему списки предпочтительнее?
17. Обоснуйте выбор между реализацией алгоритма Дейкстры на массиве $O(n^2)$ и на двоичной куче $O((n+m)\log n)$.
18. Докажите теорему о том, что высота АВЛ-дерева не превышает $1.44 * \log_2(n)$.
19. Сравните АВЛ и красно-черные деревья: какое из них обеспечивает более жесткую балансировку, а какое — более быстрые вставки/удаления и почему?
20. Обоснуйте преимущества B+-деревьев (классические B-деревья) для баз данных, особенно в контексте блочного ввода-вывода.
21. Как выбор параметра t (порядка/минимальной степени) B-дерева влияет на производительность в системах с жесткими дисками (HDD/SSD)?
22. Для структуры данных Bore докажите, что высота дерева остается $O(\log n)$ при использовании АВЛ-балансировки.

Старые вопросы (не актуально)

Вопрос №1

Сделайте сравнительный анализ O , Ω и Θ -нотаций, включая их формальные математические определения. Объясните, почему в практике программирования и при проектировании систем наибольшее распространение получила O -нотация, несмотря на её «пессимистичность».

Проиллюстрируйте свой ответ примером алгоритма, для которого O , Ω и Θ -оценки различны, и объясните, какие практические выводы для разработчика из этого следуют.

Вопрос №2

Сравните три основных метода амортизационного анализа (агрегирования, бухгалтерский, потенциалов) на примере одной и той же структуры данных (на усмотрение студента). Какой из методов, по вашему мнению, является наиболее универсальным и мощным инструментом доказательства, а какой — наиболее интуитивно понятным для программиста? Обоснуйте свою точку зрения.

Вопрос №3

На примере стека, очереди и дека раскройте практическое различие между абстрактным типом данных (АТД) и конкретной структурой данных. Почему связный список считается конкретной структурой данных, а не АТД? Проанализируйте, как выбор между реализацией на массиве (кольцевом буфере) и на связном списке влияет на: Амортизированную сложность операций, эффективность использования памяти и локальность данных и кэш-эффективность.

Вопрос №4

Развернутый связный список представляет собой гибридную структуру, сочетающую преимущества массивов и связных списков. Объясните, почему на практике выбирают фиксированные значения $k = 16-64$, а не вычисляют $\sqrt{n}$ динамически. Какие архитектурные особенности современных процессоров влияют на этот выбор? Предложите модификацию развернутого списка, которая сохраняла бы асимптотику $O(\sqrt{n})$ для операций, но улучшала производительность для паттерна доступа «частое добавление в начало». Обоснуйте эффективность вашего решения.

Вопрос №5

Проведите детальный сравнительный анализ временной и пространственной сложности для массивов, односвязных, двусвязных списков и развернутых списков. Для каждой структуры данных: Постройте таблицу сложности операций (доступ по индексу, поиск, вставка/удаление в начало/середину/конец) с указанием лучшего, худшего и среднего случаев. На основе анализа предложите оптимальную структуру данных для каждого из сценариев: а) Частые случайные доступы по индексу, редкие модификации б) Частые вставки/удаления в начале, последовательный обход в) Непредсказуемый паттерн доступа с преобладанием операций в середине структуры. Для сценария (в) обоснуйте, может ли XOR-связный список быть практичной альтернативой, учитывая его особенности и ограничения.

Вопрос №6

На примере сортировки Шелла и TimSort представьте анализ сложности сортировки Шелла. Анализ оптимальных последовательностей смещений и их асимптотической сложности. Объяснение, почему последовательность Шелла дает $O(n^2)$ в худшем случае. Для TimSort дайте строгое обоснование выбора параметра `minrun` и его оптимизации в диапазоне $[32, 64]$. Алгоритма балансировки слияния подмассивов с поддержанием инвариантов стека.

Доказательства сложности $O(n \log n)$ с учетом режима галопа. Сравните практическую эффективность этих алгоритмов для различных типов данных

Вопрос №7

Для несравнительных методов сортировки детально опишите алгоритм устойчивой сортировки подсчетом для сложных объектов, включая преобразование в префиксные суммы. Докажите корректность LSD-сортировки методом индукции по разрядам. Проанализируйте, при каких условиях цифровая сортировка превосходит сравнениями-based алгоритмы (comparison-based sorting), и почему это не противоречит теореме о нижней оценке

Вопрос №8

Проведите амортизационный анализ операции добавления в хеш-таблицу с цепочками при динамическом перехешировании: Объясните, почему порог перехеширования выбран как $4n/3$. Докажите, что амортизированная стоимость операции составляет $O(1)$, используя метод бухгалтерского учета.

Вопрос №9

Проведите сравнительный анализ алгоритмов DFS и BFS, включая математическую модель работы каждого алгоритма. Анализ временной и пространственной сложности для различных представлений графа (матрица смежности vs списки смежности). Объясните, почему DFS естественным образом использует стек, а BFS — очередь. Объясните, почему BFS гарантированно находит кратчайший путь в невзвешенных графах.

Вопрос №10

Дайте обоснование корректности алгоритма Дейкстры, доказательство того, что алгоритм не работает с отрицательными весами (приведите контрпример). Докажите, что наивная реализация с массивом имеет сложность $O(n^2 + m)$. Обоснуйте, почему использование двоичной кучи дает $O((n + m) \log n)$

Вопрос №11

Проведите детальный сравнительный анализ AVL-деревьев, красно-черных деревьев, B-деревьев, включая формальное доказательство высоты для каждого типа деревьев (теорема о высоте AVL-дерева, лемма о чёрной высоте для красно-черных, оценка высоты B-дерева). Математический анализ сложности операций вставки, удаления и поиска с учётом различных сценариев балансировки. Доказательство корректности операций балансировки (вращения в AVL, перекрашивания в красно-черных, разбиения/слияния в B-деревьях)

Вопрос №12

Для структуры данных Rore докажите, что высота дерева остается $O(\log n)$ при использовании AVL-балансировки. Проведите амортизационный анализ операций конкатенации и разделения с учётом перебалансировки. Предложите и обоснуйте модификацию Rore, которая сохраняет персистентность при сохранении асимптотики $O(\log n)$

Вопрос №13

Для B-деревьев дайте строгое обоснование операций докажите, что инварианты B-дерева сохраняются при разбиении узлов и слиянии. Проанализируйте, как выбор параметра t влияет на производительность в системах с иерархической памятью. Обоснуйте преимущества B+-деревьев перед классическими B-деревьями для систем управления базами данных.

From:

<https://se.moevm.pro/> - МОЭВМ Вики [se.moevm.pro]

Permanent link:

https://se.moevm.pro/doku.php/courses:algorithms_structures:exam:theor_analysis

Last update:

