

## *Вопросы к коллоквиуму по дисциплине*

# **Конструирование алгоритмов и структур данных**

### **Лекция 1. Введение. Асимптотический анализ. Сортировки.**

1. Дайте определение асимптотической нотации  $O$ ,  $\Omega$ ,  $\Theta$ . В чём разница между ними?
2. Проанализируйте время работы алгоритма поиска минимума в неотсортированном массиве. Почему нижняя оценка для этой задачи –  $\Omega(n)$ ?
3. Опишите алгоритм сортировки вставками. Какова его времененная сложность в худшем, лучшем и среднем случае?
4. Опишите алгоритм сортировки слиянием. Почему его сложность всегда  $O(n \log n)$ ?
5. Что такое «инверсия в массиве»? Опишите модификацию сортировки слиянием, позволяющую подсчитать количество инверсий за  $O(n \log n)$ .
6. Сформулируйте мастер-теорему. Примените её для анализа сложности сортировки слиянием и бинарного поиска.

### **Лекция 2. Бинарная куча. Пирамидальная сортировка.**

7. Что такое «очередь с приоритетом»? Какие основные операции она должна поддерживать?
8. Опишите структуру данных «бинарная куча» для минимума. Какое свойство она должна поддерживать?
9. Объясните алгоритмы операций вставка и извлечения минимума в бинарной куче. Почему их время работы –  $O(\log n)$ ?
10. Опишите алгоритм пирамидальной сортировки. Докажите, что его сложность равна  $O(n \log n)$ .
11. Как работает алгоритм построения кучи из неупорядоченного массива? Почему его сложность –  $O(n)$ , а не  $O(n \log n)$ ?

### **Лекция 3: Бинарный поиск. Порядковые статистики.**

12. Опишите алгоритм бинарного поиска в отсортированном массиве. Какова его сложность?
13. Что такое нижняя граница и верхняя граница элемента? Как их реализовать с помощью бинарного поиска?
14. В чём заключается идея «бинарного поиска по ответу»? Приведите пример задачи, где он применяется.
15. Опишите рандомизированный алгоритм быстрого выбора для нахождения  $k$ -й порядковой статистики. Какова его средняя времененная сложность?
16. Опишите идею детерминированного алгоритма медиана медиан. Почему его время работы в худшем случае –  $O(n)$ ?

#### **Лекция 4. Амортизированный анализ. Стек, очередь, дек.**

17. Что такое «амортизированная сложность»? В чём её отличие от сложности в худшем случае?
18. Объясните на примере динамического массива (стека на списке) методы агрегирования, бухгалтерского учёта и потенциалов для амортизационного анализа.
19. Как реализовать очередь с помощью двух стеков? Проведите амортизированный анализ операций enqueue и dequeue.
20. Что такое «кольцевой буфер» и как он решает проблему наивной реализации очереди на массиве?

#### **Лекция 5. Система непересекающихся множеств.**

21. Какие две основные операции поддерживает система непересекающихся множеств? Приведите примеры задач, где она применяется.
22. Опишите наивную реализацию системы непересекающихся множеств и её недостатки.
23. Что такое «ранговая эвристика» и «эвристика сжатия путей»? Как они улучшают асимптотику?
24. Какова амортизированная временная сложность операции в системе непересекающихся множеств с использованием обеих эвристик? Что такое функция Аккермана и её обратная функция?

#### **Лекция 6-7. Динамическое программирование.**

25. Сформулируйте задачу о рюкзаке. Приведите рекуррентное соотношение и объясните его.
26. Как можно оптимизировать решение задачи о рюкзаке по памяти, используя одномерный массив? Почему внутренний цикл при этом нужно проходить в обратном порядке?
27. Как восстановить набор предметов, входящих в решение задачи о рюкзаке, по заполненной таблице динамического программирования?
28. Чем решение задачи о рюкзаке с неограниченным количеством предметов отличается от классического 0/1 случая?
29. Что такое «динамическое программирование по подмножествам»? Приведите пример задачи, например, коммивояжёра.
30. Что такое «динамическое программирование по профилю»? Приведите пример задачи, например, замощение доминошками.

### **Лекция 8-9. Дерево отрезков. Дерево Фенвика.**

31. Что такое «дерево отрезков»? Какие операции и с какой сложностью оно поддерживает?
32. Опишите структуру дерева отрезков и принцип его хранения в массиве.
33. Как в дереве отрезков реализованы операции запроса на отрезке и обновления элемента?
34. В чём заключается идея «отложенных обновлений» в дереве отрезков? В каких случаях она необходима?
35. Что такое «дерево Фенвика»? Какие операции оно поддерживает и в чём его преимущества перед деревом отрезков?
36. Объясните, как работает операция вычисления суммы на префикссе в дереве Фенвика, используя операцию  $i$  &  $-i$ .

### **Лекция 10. Разреженная таблица.**

37. Для каких задач предназначена разреженная таблица? Каковы её ограничения?
38. Опишите процесс предобработки и выполнения запроса в разреженной таблице для задачи поиска минимума на отрезке.
39. Почему операция запроса в разреженной таблице выполняется за  $O(1)$ , но при этом неприменима для обновлений?

### **Лекция 11. Сбалансированные деревья поиска.**

40. Что такое «сбалансированное дерево поиска» и зачем оно нужно? В чём проблема несбалансированного двоичного дерева поиска?
41. Опишите инвариант AVL-дерева. Какие повороты используются для поддержания его сбалансированности?
42. Опишите основную идею Splay-дерева. Что такое операция подъёма к корню и какие шаги она включает?
43. Сравните АВЛ-дерево и Сплей-дерево. В чём преимущества и недостатки каждого с точки зрения времени работы, констант и областей применения?

И.о. зав. кафедрой ВТ

Т. А. Приходько

Составитель: ст. преп. КВТ

В. И. Шиян