

Яндекс Образование
Курс по олимпиадному программированию
Параллель С

25.07.2023

Содержание

1	Оценка времени работы программы	3
2	Бинарный Поиск #1	3
3	Теория чисел	3
4	С++	4
5	Динамическое программирование #1	4
6	Динамическое программирование #2	4
7	Динамическое программирование #3	5
8	Структуры данных #1	5
9	Графы #1	5
10	DFS #1	6
11	BFS	6
12	Жадные алгоритмы	6
13	Коллоквиум	6
14	Интерактивные задачи	6
15	Корневая декомпозиция	6
16	Битовые операции	6

17 Куча	7
18 Кратчайшие пути	7
19 Бинарный Поиск #2	7
20 Два указателя	7
21 Работа с памятью	8
22 DFS #1	8
23 Комбинаторика #1	8
24 Комбинаторика #2	8
25 Работа с случайными данными	8
26 Геометрия - примитивы	9
27 Геометрия - прямые	9
28 Игры	9
29 Резерв	9
30 Зачет	9

1 Оценка времени работы программы

1. O-нотация. Математическое определение. Замечание о том, что для конкретных ограничений алгоритм, более быстрый асимптотически, не всегда действительно быстрее.
Сколько операций в секунду способен обрабатывать компьютер (порядка 10^8).
2. Навык по ограничениям в задаче понять, решения каких асимптотик найдут, а какие нет.
3. Квадратичные сортировки: пузырьком, выбором, вставками.
Сортировка подсчетом.

2 Бинарный Поиск #1

1. Бинарный поиск. Понимание того, что для бинарного поиска нужны:
 - a) Монотонная функция.
 - b) Инвариант для границ бинарного поиска.
2. Вещественный бинарный поиск. Замечание о том, что лучше всего устанавливать константное число итераций.
3. Утверждение о том, что непрерывная функция, принимающая в одной точке неположительное значение, а в другой – неотрицательное, имеет точку, в которой принимает нулевое значение. Применение бинарного поиска для поиска этой точки.

3 Теория чисел

1. НОД и НОК.
2. Алгоритм Евклида.
3. Расширенный алгоритм Евклида.
4. Нахождение всех делителей числа за $O(\sqrt{n})$.
5. Факторизация за $O(\sqrt{n})$.
6. Быстрое возведение в степень.
7. Решето Эратосфена за $O(n \log \log n)$.
8. Применение решета Эратосфена для факторизации чисел за $O(\log n)$ с предсчетом за $O(n)$.

4 C++

1. *ios :: sync_with_stdio(false);*
cin.tie(nullptr);
cout.tie(nullptr);
2. *cppreference*
3. Работа с файлами:
freopen
ifstream, ofstream
Бонус: *sstream*
4. *gcd(C++17 - gcd, lcm)*
min_element, max_element, nth_element
merge, sort, stable_sort
fill, copy, (C : memset, memcpy)
reverse, rotate
unique
lower_bound, upper_bound, binary_search
next_permutation, prev_permutation
partial_sum
5. Компараторы-функторы.
6. Вектор.
7. *priority_queue*.

5 Динамическое программирование #1

1. Основы ДП. Кузнечик + черепашка (подсчет количества путей, подсчет минимального/максимального пути).
Формализуем сказанное: что нужно для ДП?
 - а) Какое состояние ДП? Что такое ответ для состояния (целевая функция)?
 - б) Какие начальные значения?
 - в) Как пересчитывать значения?
 - г) Какой должен быть порядок обхода, чтобы все значения, требуемые при пересчете, уже были подсчитаны?
 - е) Как получить ответ?

6 Динамическое программирование #2

1. Восстановление ответа: через массив динамики и через массив предков.

2. Рюкзак. Взвешенный/невзвешенный. Число предметов ограничено/не ограничено.
Реализация на двумерном массиве. Реализация на одномерном массиве (храним лишь текущий слой): замечание о том, что решения для ограниченного и неограниченного количества предметов отличаются направлением цикла for.
3. Динамика по префиксам. Примеры:
 - а) НВП.
 - б) Количество строк заданной длины из какого-то ограниченного алфавита, не содержащих другую заданную строку в качестве подпоследовательности.

7 Динамическое программирование #3

1. Рекурсия. Примеры:
 - а) Вычисление чисел Фибоначчи с помощью рекурсии.
 - б) Вывод перевернутого массива с помощью рекурсии.
 - в) Задача о ханойских башнях.
2. Как устроена рекурсия на компьютере. Что такое стек рекурсии.
3. Как сделать, чтобы глубокая рекурсия работала у вас? (ulimit -s unlimited на Linux/MacOS, sys.setrecursionlimit для Python).
4. Как переделать рекурсию на stack.
5. Ленивая динамика: как удобно писать. Примеры задач, где без неё никак. Провести аналогии с DFS.
6. Динамика по состоянию последнего элемента: количество последовательностей из 0 и 1 без двух 0 подряд.

8 Структуры данных #1

1. Стек, очередь, дек.

9 Графы #1

1. Неориентированные графы.
Основные определения: вершина, ребро, путь, цикл, простой путь, простой цикл.
Изображение графов. Примеры.
Что можно представлять в виде графов? (Например, отношение дружбы.)
Ориентированные графы.

Взвешенные графы.

Как хранить графы?

- a) Матрица смежности.
- b) Списки смежности.

2. Деревья. Определение дерева. Эквивалентные определения дерева. Подвешенные деревья.

Как реализовывать DFS на деревьях без массива used?

10 DFS #1

1. Базовый DFS. Примеры:
 - a) Проверка графа на связность.
 - b) Выделение всех компонент связности.
2. Проверка на двудольность.

11 BFS

1. BFS.
2. 0-k BFS (в частности, 0-1 BFS = deque).

12 Жадные алгоритмы

13 Коллоквиум

14 Интерактивные задачи

1. Интерактивные задачи. flush.

15 Корневая декомпозиция

1. Корневая декомпозиция массива для нахождения функции на отрезке.

16 Битовые операции

1. Битовые операции: $\&$, $|$, \oplus .
2. Представление чисел в памяти
3. Перебор всех подмасок данной маски.
4. Оценка 3^n на суммарное количество подмасок для всех масок, меньших 2^n .

17 Куча

1. Куча: операции `sift_down`, `sift_up`. Как реализовывать `priority_queue`: извлечение минимума и добавление. Удаление/изменение веса произвольного элемента. `Heap sort`.

18 Кратчайшие пути

1. Дейкстра: базовая и на сети (`priority_queue`!).
2. Форд-Беллман.
3. Флойд.
4. Детекция циклов отрицательного веса.

19 Бинарный Поиск #2

1. Бинарный поиск по ответу. Стандартные задачи с каким-нибудь процессом и просьбой минимизировать что-нибудь. Смежные с другими темами примеры: бинпоиск по ответу в графах.
2. Представление бинпоиска как поиска по степеням двойки: сначала пробуем установить самый старший бит в 1, и т.д.
Замечание о том, что можно делать такой поиск и в других системах счисления.
3. Тернарный поиск. Примеры функций, к которым можно применять и к которым нельзя. Замечание о том, что, если функция не меняет своего значение лишь в точках экстремума, к ней все же можно применить тернарный поиск.
4. Бинарный поиск по производной.
5. Тернарный поиск с золотым сечением.

20 Два указателя

1. Два указателя. Примеры:
 - a) `lower_bound` для двух отсортированных массивов.
 - b) Поиск подотрезка максимальной длины с количеством различных $\leq k$.
2. Поиск подотрезка с максимальной суммой. Два способа:
 - c) Считаем префиксные суммы и поддерживаем минимальную.
 - d) Делаем `cur = max(cur + a[i], a[i])`.

21 Работа с памятью

1. Как считать, сколько памяти будет использовать программа: что такое бит, (кило/мега/гига)байт, сколько байт обычно занимает каждый из примитивных типов.
2. Сколько вмещает в себя каждый тип данных, знаковые-беззнаковые типы, про точность вещественных чисел
3. Merge sort. Реализация красивая с $O(n \log n)$ доп. памяти. Реализация на подотрезках с n доп. памяти.
4. Применение merge sort для поиска количества инверсий.

22 DFS #1

1. DFS: время входа-выхода и топсорт.
2. DFS: конденсация.

23 Комбинаторика #1

1. Правила суммы и произведения.
2. Перестановки. Факториал как количество перестановок длины n .
3. Треугольник Паскаля.
4. Бином Ньютона.

24 Комбинаторика #2

1. Поиск числа сложных комбинаторных объектов через динамику: разные виды ПСП, разные разбиения на слагаемые, разные последовательности со странными свойствами (убывающие, возрастающие).
2. Генерация всех комбинаторных объектов в лексикографическом порядке: перестановок, сочетаний, размещений (с повторениями и без), ПСП, убывающие-возрастающие последовательности, разбиения на слагаемые (разные виды).

25 Работа с случайными данными

1. Что такое ожидаемое время работы?

2. Quick sort. Реализация со взятием центрального элемента в качестве: утверждение о том, что легко построить тест, на котором это $O(n^2)$. Реализация со взятием случайного элемента в качестве опорного: утверждение о том, что ожидаемое время работы составляет $O(n \log n)$.
3. Как правильно писать рандом.

26 Геометрия - примитивы

1. Структура вектор.
2. Скалярное и векторное (псевдоскалярное) произведения.
3. Площадь многоугольника.
4. Расстояние от точки до прямой.
5. Проекция точки на прямую.

27 Геометрия - прямые

1. Общее уравнение прямой. Переход от него к двум точкам и наоборот. Вектор нормали.
2. Пересечение прямых.
3. Пересечение прямой и окружности.
4. Пересечение двух окружностей.
5. Поиск касательных к окружности.

28 Игры

1. Выигрышные, проигрышные состояния.
2. Ретро-анализ.

29 Резерв

30 Зачет