

Яндекс Образование
Курс по олимпиадному программированию
Параллель А

25.07.2023

Содержание

1	Теория чисел	3
2	Оптимизации ДП	4
3	Невзвешенные графы	4
4	Корневая декомпозиция	5
5	Структуры данных на отрезках	5
6	linux + C++	5
7	Взвешенные графы	7
8	Геометрия #1	7
9	Строки #1	8
10	Прокачиваем дерево отрезков	9
11	Структуры на деревьях	9
12	Идейный семинар	9
13	FFT	9
14	Зачет	9
15	Развлекательный контекст	9
16	Потоки	9

17 Теорвер и линал	10
18 Строки #2	10
19 Битовые оптимизации и переборы	11
20 Игры	12
21 Неточки	12
22 Математика #228	12
23 Метод потенциалов	13
24 Графы	13
25 Паросочетания	13
26 Симплекс	13
27 Haskell	13

1 Теория чисел

1. Асимптотики
2. Алгоритм Евклида.
3. Бинарный алгоритм Евклида.
4. 2 алгоритма Евклида для длинных чисел.
5. Утверждение о том, что алгоритм Евклида для n чисел, не превосходящих C , работает за $O(n + \log C)$.
6. Утверждение о том, что НОД, уменьшаясь, уменьшается хотя бы в 2 раза.
7. Расширенный алгоритм Евклида.
8. Обратный по модулю. Критерий существования.
9. Применение расширенного алгоритма Евклида для поиска обратного по произвольному модулю.
10. Поиск обратного по произвольному модулю в 2 строчки.
11. Функция Эйлера и ее мультипликативность.
12. Вычисление функции Эйлера за $O(\sqrt{n})$.
13. Теорема Эйлера, малая теорема Ферма и их применение для поиска обратных по модулю.
14. Нерекурсивное бинарное возведение в степень.
15. Утверждение о том, что количество простых чисел, не превосходящих n , – это $O(n/\log n)$.
16. Три метода для поиска обратных ко всем остаткам по простому модулю за $O(p)$.
17. Китайская теорема об остатках.
18. Решето Эратосфена за $O(n)$.
19. Применение решета Эратосфена для подсчета мультипликативных функций для всех натуральных чисел, не превосходящих n , за $O(n)$.
20. Применение решета Эратосфена для факторизации чисел за $O(\log n)$ с предпосчетом за $O(n)$.
21. Количество делителей – субполиномиальная функция, практическая оценка $O(n^{1/3})$, алгоритм для поиска сверхсоставных чисел, таблицы сверхсоставных для практики.

22. Замечание о том, что количество различных простых делителей крайне мало.
23. Задача дискретного логарифмирования и ее решение за $O(\sqrt{n})$ методом meet-in-the-middle.
24. Нижняя и верхняя оценка на частичные суммы гармонического ряда..

2 Оптимизации ДП

1. Использование вспомогательной динамики для пересчета значения в вершине.
2. Задача о редукции дерева: удалить из дерева минимальное количество ребер так, чтобы одна из компонент была размера ровно k .
3. Оптимизация разделяй-и-властвуй.
4. Оптимизация Кнута.
5. Convex hull trick.
6. Convex hull trick в дереве отрезков.
7. Дерево Li Chao.
8. Лямбда-оптимизация.
9. MOD^2 -оптимизация.
10. Хранение только достижимых состояний.
11. Менять местами значение дп и параметр($dp[a][b] = c \rightarrow dp[a][c] = b$).
12. Функция ДП – пара.
13. Ускорение ДП возведением матриц в степень.
14. Оптимизации возведения матриц в степень: MOD^2 , транспонирование.
15. Ускорение вычисления ДП для разных степеней предпосчетом матриц перехода для степеней двойки.

3 Невзвешенные графы

1. Эйлеров путь и цикл
2. Лемма Холла
3. Теорема Кёнига
4. Семинар

4 Корневая декомпозиция

1. Нахождение количества треугольников в графе за $O(E\sqrt{E})$.
2. Корневая для задачи dynamic records.
3. Корневая в задачах на строки: разбиение строк на короткие и длинные.
4. Корневая на графах: тяжелые и легкие вершины.
5. Решение задачи о рюкзаке за $O(S\sqrt{S})$
6. Алгоритм Мо.
7. Алгоритм Мо на дереве (эйлеровом обходе).
8. Алгоритм Мо + корневая как структура для Мо.
9. 3D Мо.
10. Мо онлайн.
11. Split-rebuild и split-merge в корневой.
12. Корневая декомпозиция по запросам.

5 Структуры данных на отрезках

1. Фенвик
2. Оптимальная нумерация вершин дерева отрезков
3. Дерево отрезков снизу
4. Неявное ДО
5. merge-sort tree
6. 2D ДО
7. disjoint sparse table

6 linux + C++

1.

```
ios :: sync_with_stdio(false);
cin.tie(nullptr);
cout.tie(nullptr);
```
2. cppreference

3. `scanf printf setprecision` для отформатированного вывода
4. Работа с файлами:
`freopen`
`ifstream, ofstream`
 Бонус: `sstream`
5. `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`
6. `pb_ds : ordered_set, gp_hash_table`, быстрый `priority_queue`
7. Лямбда-функции.
8. Компараторы-функторы.
9. range-based for
 Structured binding declaration
10. `#pragma`
11. `define ifdef`
12. `priority_queue::set`
13. `initializer_list`
14. `swap`, его время работы на стандартных контейнерах ($O(1)$)
15. `mt19937, mt19937_64`
`random_device`
`shuffle`
16. `templates, template specialization`
17. Классы, наследование, виртуальные функции
18. AVX
19. Написание стресс-тестов с помощью `python`
 Написание стресс-тестов с помощью `bash`
 Написание простых генераторов
20. `-fsanitize = address, undefined, bounds - g`
`valgrind`
`gdb`
21. `Makefile`

7 Взвешенные графы

1. Минимальные остовы.
2. Алгоритм Прима.
3. Алгоритм Крускала.
4. Алгоритм Борувки.
5. Поиск мостов онлайн.
6. Проведение ребер на отрезке.
7. СММ.

8 Геометрия #1

1. Расстояние от точки до прямой.
2. Пересечение прямых.
3. Пересечение прямой и окружности.
4. Пересечение двух окружностей.
5. Поиск касательных к окружности.
6. Проверка на принадлежность точки многоугольнику за $O(n)$. Два способа: сумма углов и луч.
7. Алгоритм Джарвиса.
8. Алгоритм Грэхема.
9. Алгоритм Эндрю.
10. Алгоритм Чана.
11. Поиск пары пересекающихся отрезков за $O(n \log n)$.
12. Локализация точки в выпуклом многоугольнике за $O(\log n)$ на запрос.
13. Поиск касательных из точки к выпуклому многоугольнику за $O(\log n)$ на запрос.
14. Пересечение прямой с выпуклым многоугольником за $O(\log n)$ на запрос.
15. Локализация точки в невыпуклом многоугольнике за $O(\log n)$ на запрос.
16. Поиск двух ближайших точек в 2D.

17. Поиск двух ближайших точек в 3D.
18. Вращающийся scanline. Запросы количества точек в полуплоскости. $O(n^2 + q \log n)$. Возможность применения корневой.
19. Сумма Минковского и ее применения.
20. Квадродерево.
21. Проецирование на случайную прямую.
22. Пересечение полуплоскостей за $O(n^2)$.
23. Проверка на непустоту пересечения полуплоскостей за $O(n)$.
24. Поиск минимальной покрывающей окружности за $O(n)$.
25. Пересечение полуплоскостей за $O(n \log n)$.
26. Триангуляция методом отрезания ушей за $O(n^2)$
27. Диаграмма Вороного за $O(n^2)$.
28. Диаграмма Вороного за $O(n \log n)$.
29. Триангуляция Делоне.

9 Строки #1

1. Хеш-таблицы. Открытая и закрытая адресации.
2. Хеши.
3. Хеши мультимножеств.
4. Проверка корневых деревьев на изоморфизм.
5. Парадокс дней рождения.
6. Поиск тандемных повторов
7. Алгоритм Манакера
8. Бор. Способы хранения: *map*, *unordered_map* (один большой), массив, вектор.
9. Сжатый бор. Оценка на глубину $O(\sqrt{S})$.
10. Ахо-Корасик.
11. Суффиксный массив за $O(n \log n)$.
12. Подсчет LCP за $O(n)$ в суфмассе.
13. Суффиксный массив за $O(n)$.
14. ALCS

10 Прокачиваем дерево отрезков

1. Персистентность
2. DCP
3. Li chao

11 Структуры на деревьях

1. Двоичные подъемы:
Аналогия с бинарным поиском.
Задача LA.
Задача LCA.
2. Эйлеров обход дерева.
3. Сжатые деревья.
4. Heavy-light decomposition. Объединение HLD и эйлерова обхода.
5. Кеширование префиксов в HLD.
6. Продвинутое применение эйлерова обхода.
7. Ladder decomposition и k -th ancestor. Построение за $O(n \log n)$, ответ на запрос за $O(1)$.
8. Ladder decomposition и k -th ancestor. Построение за $O(n)$, ответ на запрос за $O(1)$.
9. Centroid decomposition.
10. Оптимизация меньшее к большему (вариация – отрезай меньшее).
11. Переливания в дереве за $O(h) \implies O(n)$.
12. Обходим маленькие поддеревья дважды.

12 Идеальный семинар

13 FFT

14 Зачет

15 Развлекательный констест

16 Потoki

1. Теорема Форда-Фалкерсона и соответствующий алгоритм.

2. Алгоритм Эдмондса-Карпа.
3. Масштабирование для Форда-Фалкерсона и Эдмондса-Карпа.
4. Декомпозиция потока за $O(E^2)$ и за $O(VE)$.
5. LR-потоки.
6. Алгоритм Диница.
7. Масштабирование для Диница.
8. Оценки Карзанова.
9. Поиск величины максимального потока в планарном графе.
10. Стоимостные потоки. Форд-Беллман на очереди, Дейкстра с потенциалами.

17 Теорвер и линал

1. Определение вероятности. Подсчет простых вероятностей.
2. Условные вероятности.
3. Доказательство парадокса дня рождений для 23 человек
4. Матожидание. Его линейность.
5. Алгоритм Карацубы.
6. Введение в комплексные числа. Применение комплексных чисел в геометрии.
7. Быстрое преобразование Фурье.
8. Теоретико-числовое преобразование Фурье.
9. Применение Фурье к решению задач.

18 Строки #2

1. Дерево палиндромов.
2. Суффиксный автомат.
3. Суффиксное дерево.

19 Битовые оптимизации и переборы

1. Динамика по изломанному профилю. Задача о количестве замощений доминошками.
2. Решение задачи суммы в подмножестве за $O(2^n)$.
3. Поиск гамильтонова пути за $O(2^n \cdot n^2)$.
4. Нахождение старшего ненулевого бита для всех чисел, меньших 2^n , за $O(2^n)$.
5. Нахождение младшего ненулевого бита для всех чисел, меньших 2^n , за $O(2^n)$.
6. Оценка $2^{n-1} \cdot n$ на суммарное количество ненулевых битов для всех чисел, меньших 2^n .
7. Перебор всех подмасок данной маски.
8. Оценка 3^n на суммарное количество подмасок для всех масок, меньших 2^n .
9. Замечание о том, что *vector* < bool > применяет битовое сжатие. Размышления о том, что это может как негативно, так и положительно влиять на скорость выполнения программ.
10. *bitset* и его простейшие применения. Рассказ о *_Find_first* *_Find_next*.
11. Транзитивное замыкание Флойдом за $O(n^3/w)$.
12. Способ представления знаковых чисел (дополнительный код).
13. Нахождение количество ненулевых битов в числе: предподсчет + деление на части, *builtin_popcount*, *builtin_popcountll*, `#pragma GCC optimize("popcnt")` vs *precalc*.
14. Применение многомерных префиксных сумм для решения задачи суммы по подмаскам за $O(2^n * n)$.
15. Поиск гамильтонова пути за $O(2^n \cdot n * \lceil n/w \rceil)$ ($= O(2^n * n)$ для актуальных значений n).
16. Поиск максимальной клики за $O(2^{(n/2)})$ методом meet-in-the-middle.
17. Битовые свертки: *and*, *or*.
18. Битовые свертки: *xor*
19. Меморизация в задаче о клике (даёт результат не хуже meet-in-the-middle)
20. Предподсчёт ответа для маленьких задач

21. Состояния перебора — это дерево
22. Оптимизации обхода большого дерева с добавлением рандома
23. iterative deepening (Увеличиваем максимальную длину пути в дереве перебора на 1)
24. Meet-in-the-middle

20 Игры

1. Выигрышные, проигрышные состояния.
2. Теория Шпрага-Гранди.
3. Теория Смита (игры с ничейными состояниями)
4. Ретро-анализ.
5. Альфа-бета отсечение.
6. Разрешающие деревья: адаптивная и неадаптивная модели.
Задача об угадывании числа, задача о нахождении самого большого числа, задача сортировки сравнениями, задача о проверке на связность.

21 Неточки

1. Алгоритм отжига.
2. Генетический алгоритм.
3. Генетический алгоритм. Монте-Карло.
4. Генетический алгоритм. Поточковый алгоритм поиска количества различных.

22 Математика #228

1. Теорема Люка.
2. Функция Мёбиуса.
3. Свертка Дирихле.
4. Первообразный корень и его поиск.
5. Поиск квадратного корня по простому модулю за $O(\log p)$.
6. Тройчная сбалансированная система счисления.

7. Hockey-stick identity и подсчет различных сумм в треугольнике Паскаля.

23 Метод потенциалов

1. Что такое амортизированное время работы?
2. Метод потенциалов.
3. Dynamic connectivity problem online за $O(n \log^2 n)$.
4. Splay-дерево.
5. Link-cut.
6. ДО + (ДД/Splay) для dynamic records.
7. Биномиальная куча.
8. Фибоначчиева куча.

24 Графы

1. 2 китайца
2. Доминаторы
3. Остов за линию
4. push relabel

25 Паросочетания

1. Венгерский алгоритм.
2. Алгоритм сжатия соцветий (поиск максимальных паросочетаний в произвольных графах).
3. Алгоритм Збана
4. Взвешенное паросочетание в произвольном графе.

26 Симплекс

27 Haskell