

ЛКШ.2014.Июль, параллель А', теорзачёт

Дополнительные вопросы по языку C++

1. (3) C++ : bitset, nth_element, sort, stable_sort, lower_bound, merge
2. (3) C++ : реализация своего компаратора для sort, merge, lower_bound, set
3. (3) C++ : указатель, ссылка, передача параметров
4. (4) C++ : скорость работы vector<int>, overload operator new, operator delete
5. (3) C++ : struct, class, public, private, конструктор, list initialization, деструктор
6. (3) C++ : финализатор, как деструктор
7. (3) C++ : быстрое чтение данных, random_shuffle
8. (3) C++ : static, inline

Основные вопросы по пройденным темам

1. (3) RMQ : Дерево отрезков, Sparse Table. Реализация дерева отрезков снизу.
2. (3) LCA : Двоичные подъемы, две реализации, минимум на пути дерева. Проверка за $O(1)$, является ли вершина предком.
3. (4) RMQ & LCA : Эйлеров обход. Построение декартова дерева.
4. (4) LCA : Offline алгоритм Ахо-Ульмана-Тарьяна
5. (5) RMQ : Sparse Table рекурсивный. Алгоритм Фараха-Колтона-Бендера.
6. (4) ScanLine : Дерево отрезков сверху. Групповые операции: $+=$, $=$, \min , sum . Задача "площадь объединения прямоугольников". k -инверсии.
7. (3) ScanLine : Задачи. 1. Количество точек в прямоугольнике. 2. Возрастающая последовательность точек. 3. Для каждой точки узнать, сколько раз она покрыта
8. (4) Корневая : идея отложенных операций. Примеры задач: 1. permutation2 (двумерный запрос на массиве, массив меняется); 2. sum2 (числа добавляются в множество, сумма на отрезке); 3. сумма на меняющемся массиве (изменить в точке, сумма на отрезке);
9. (5) Корневая : идея нескольких отрезков, идея split & rebuild. Reverse на отрезке + двумерный запрос на массиве + присваивание на отрезке.
10. (4) Дерево отрезков сортированных массивов (оно же двумерное дерево). Примеры задач на него.
11. (4) Декартово дерево (treap). RBST. Персистентное декартово дерево
12. (3) Идея неявного ключа, reverse на отрезке + сумма на отрезке = $O(\log n)$.
13. (3) Персистентный массив. Персистентность всего мира.
14. (4) Сравнение строк хешами на равенство. Вероятности. Коллизии. Хештаблица с открытой адресацией. Решение задачи про жужжащего профессора хешами.
15. (3) Сравнение строк хешами на больше-меньше. LCP. Суффиксный массив за $O(n \log^2 n)$.
16. (3) Хештаблица на списках. LCP для всех пар (i, j) динамикой.
17. (4) Суффиксный массив за $O(n^2)$ и $O(n \log n)$ цифровой сортировкой. Алгоритм Касаи для нахождения LCP.
18. (4) Бор. Реализация. Сжатый бор. Суффиксное дерево за $O(n^2)$. Решение задач про число подстрок, про поиск строки в тексте суффиксным деревом.
19. (3) Общая подстрока k строк за $O(nk \log n)$. LCP в суффиксном массиве за $O(1)$.
20. (3) "Суффиксное дерево" \leftrightarrow "суффиксный массив и LCP" за $O(n)$.
21. (4) Укконен без суффиксных ссылок за $O(n^2)$.
22. (5) Укконен за $O(n)$ с доказательством.
23. (4) Skew Heap, сливаемые кучи. С доказательством.
24. (4) Игра на ацикличном графе, DP за $O(E)$. Ретроанализ, реализация за $O(E)$. Длина игры. Длина игры в поддавки. Ретроанализ на графе из двух цветов.
25. (5) Гранди с доказательством. Эквивалентность игр.
26. (3) Задачи на гранди: ним, ним с делением попалам, карлсон, пешки, скамейка, hacking bush
27. (3) matching, cover, independent set, cover \leftrightarrow independent set, $\min \text{cover} \geq \max \text{matching}$
28. (4) Лемма про дополняющую цепь. dfs для поиска пути. Алгоритм Куна. Доказательство. Жадная инициализация паросочетания.
29. (3) Контролирующее и независимое множества в двудольном графе по максимальному паросочетанию за $O(E)$.
30. (3) Задачи на паросочетание: покрытие грида доминошками, покрытие ацикличного графа путями, максимальные клика и антиклика.
31. (4) Рандомизированный алгоритм для паросочетания в произвольном графе. Стабильное паросочетание с доказательством корректности.
32. (3) Поток в графе, разрез в графе. $\min \text{cut} \geq \max \text{flow}$. Ориентированный граф. Неориентированный граф. Вершинный поток. Декомпозиция потока. Паросочетание в двудольном графе через поток.

33. (4) Теорема и алгоритм Форда-Фалкерсона. Задача про k непересекающихся путей за $O(kE)$.
34. (5) Алгоритмы Эдмондса-Карпа и Scaling с доказательствами.
35. (4) Несколько истоков, стоков. Избытки, недостатки. [L,R]-циркуляция. [L,R]-поток. [L,R]-max-поток.
36. (3) Задачи: выделение k -регулярного графа из двудольного, футбольный турнир, восстановление матрицы
37. (5) Лемма Холла. Существование совершенного паросочетания в регулярном двудольном графе. Разбиение регулярного графа на паросочетание. Покраска ребер регулярного графа.
38. (5) MinCost Flow. Построение графа, алгоритм поиска, доказательство корректности.
39. (4) Дейкстра и потенциалы. Взвешенное паросочетание в двудольном графе за $O(V^3)$.
40. (3) Форд-Беллман с очередью. Реализация, доказательство корректности. Задача про ориентацию графа.
41. (3) Геометрия. Погрешность. Скалярное и векторное произведение. Расстояние от круга до квадрата.
42. (3) Выпуклая оболочка за $O(n \log n)$. Проверка, лежит ли точка в многоугольнике за $O(n)$. Ориентация многоугольника по часовой стрелке.
43. (4) Проверить, есть ли пересечение отрезков за $O(n \log n)$. Найти все пересечения отрезков за $O((n+k) \log n)$.
44. (4) Расстояние между двумя выпуклыми непересекающимися многоугольниками за $O(n)$. Максимальное расстояние между точками в 2D за $O(\text{ConvexHull} + n)$.
45. (4) Треугольник и четырехугольник максимальной площади за $O(n^2)$. Количество четверок точек, образующих выпуклый четырехугольник за $O(n^2 \log n)$.
46. (3) Задачи на сортировку : 1. Унести в рюкзак максимальное число вещей. 2. Выполнить как можно больше заказов, у каждого есть $time_i$. 3. Выполнить все заказы, у каждого есть $deadline_i$ и $time_i$. 4. Непрерывный рюкзак с весами и стоимостями. 5. Есть работы, t_i , fee_i , время выполнения — T_i , минимизировать $\sum fee_i \cdot T_i$.
47. (3) Задачи на сортировку : 1. Задача про 2 станка 2. Построить башенку из всех коробок, у каждой есть масса m_i и прочность s_i . 3. Сконкатенировать все строки так, чтобы результат был минимален.
48. (5) Решения задачи про коробки (построить башенку из максимального числа коробок, у каждой есть масса m_i и прочность w_i). Два решения динамикой за $O(n^2)$. Два решения за $O(n \log n)$ (деревом отрезков, set-ом). Доказательство корректности всех решений кроме решения с set-ом.
49. (3) Жадность и перебор на примере задачи о гамильтоновом пути: перебор с отсечениями, с запоминанием; перебор с эвристикой минимальной степени; жадность с `random_shuffle`.

Алгоритм сдачи экзамена

1. Получить вопрос на (4) или два вопроса на (3) по выбору преподавателя
2. Ответить, если успех, у вас гарантирована отметка 4-
3. Получить задачу на 4, если ответить, гарантирована оценка 4+
4. Получить билет на (5) **или задачу на 5** (одно из двух, по обоюдному желанию), если ответить, гарантирована оценка 5
5. Получить задачу на 5+, и идите решайте её...
6. Если не ответить задачу на 4, можно взять вторую и подняться до 4, третью и подняться до 4+, но выше 4+ уже не подняться.
7. Если не ответить билет/задачу на 5, можно получить другую задачу на 5 и если вы ее решите, оценка 5-.
8. Если не ответить билет на (4), можно получить билет на (3), ответив его получить 3 и набирать задачи на 4 до оценки 4-.
9. Если не ответить билет на (3), можно взять другой билет на (3) и получить 3-.

Дополнительное правило: к вам в любой момент может подойти преподаватель и дополнительными простыми вопросами или понизить вашу оценку на треть балла, или, если борьба идет в районе 3-4, то даже поднять вашу оценку на треть балла.