

Вопросы к коллоквиуму по алгоритмам

СПб АУ, первый курс, осень, 2016/17 учебный год

Курсивом помечено то, что было разобрано на практике.

Динамическое программирование

1. (a) DP. Вперёд, назад, ленивая. Восстановление ответа. Пример: «калькулятор». Ациклический граф состояний. Формулировка задачи в терминах графа. Формулировка терминов «вперёд», «назад», «ленивая» через граф состояний.
2. (a) DP. Задача про путь на матрице с дырками. Min путь, max путь, кол-во путей.
3. (b) DP. Рюкзак без стоимостей за $\mathcal{O}(nW)$ времени, $\mathcal{O}(W)$ памяти с восстановлением ответа. Применение bitset. Проблемы с восстановлением ответа в случае стоимостей. *Решение рюкзака со стоимостями.*
4. (a) DP. Задача НОП. Восстановление ответа без дополнительных ссылок назад. Оптимизация памяти, если нет восстановления ответа.
5. (b) DP. Оптимизация памяти для восстановления ответа в НОП с помощью bitset.
6. (c) DP. Алгоритм Хиршберга для НОП за $\mathcal{O}(n^2)$ времени, $\mathcal{O}(n)$ памяти с восстановлением ответа. *Хиршберг для рюкзака.*
7. (a) DP. *Задача LCP.* Задача «расстояние Левенштейна».
8. (b) DP. Задача LIS (НВП). Решения за $\mathcal{O}(n^2)$ и за $\mathcal{O}(n \log n)$.
9. (b) DP. Измельчение перехода. Задача про погрузку грузов на корабль, решения за $\mathcal{O}(n^4)$, $\mathcal{O}(n^3)$, $\mathcal{O}(n^2)$.
10. (b) DP. Возведение матрицы в степень: количество путей длины k , решение линейного рекуррентного соотношения.
11. (b) DP. Два указателя на примере задачи про выбор k точек на прямой среди n данных. Без док-ва.
12. (c) DP. Доказательство $p_{k-1,n} \leq p_{k,n} \leq p_{k,n+1}$.
13. (b) DP. Решение задачи про выбор k точек на прямой среди n данных методом «разделяй и властвуй».
14. (a) DP. Подотрезки. Умножение матриц, расстановка скобок в выражении. Восстановление ответа.

Комбинаторика и подмножества

15. (b) Комбинаторика. Объект по номеру, номер по объекту. Перестановки, скобочные последовательности.
16. (a) Комбинаторика. *Подсчет числа разбиений на слагаемые: упорядоченные, неупорядоченные.*
17. (a) DP. Операции с множествами, как с n -битовыми числами. Количество бит в числе за $\mathcal{O}(2^n)$. Сумма на подмножестве за $\mathcal{O}(2^n)$. Разворот битовой записи.
18. (b) DP. Гамильтонов путь и цикл за $\mathcal{O}(2^n n)$.
19. (a) DP. *Проверка независимости всех подмножеств вершин за $\mathcal{O}(2^n)$.*
20. (b) DP. Перебор подмножеств и надмножеств. Покраска вершин графа в минимальное число цветов за $\mathcal{O}(3^n)$.
21. (c) DP. Оптимизация покраски вершин графа в минимальное число цветов с $\mathcal{O}(3^n)$ до $\mathcal{O}(2.44^n)$.
22. (a) DP. Subset Cover. Рюкзак на подмножествах.
23. (b) Meet in the Middle. Подсчет числа клик за $\mathcal{O}(2^{n/2})$ двумя способами.
24. (b) Meet in the Middle. Рюкзак за $\mathcal{O}(2^{n/2} n)$.
25. (c) DP. Динамика по скошенному профилю: количество замощений доминошками за $\mathcal{O}(2^h h w)$. Только рекурсивная реализация.

Графы и dfs

26. (a) База. Хранение графов. Матрица смежности, bitset<int>[], vector<int>[], set<int>[], мультисписок.

27. (a) dfs. Def: путь, простой путь. Поиск простого пути между a и b dfs-ом за $\mathcal{O}(E)$. Def: связность, сильная связность, слабая связность, компоненты связности. Поиск компонент связности за $\mathcal{O}(V + E)$.
28. (a) dfs. Циклы. Def: цикл, контур. Поиск цикла в орграфе, в неорграфе за $\mathcal{O}(V + E)$. Проверка на двудольность.
29. (b) dfs. Поиск topsort двумя способами. Связь с динамическим программированием.
30. (b) dfs. Выделение компонент сильной связности за $\mathcal{O}(V + E)$. Конденсация графа.
31. (b) dfs. Euler. Критерий Эйлеровости. Поиск Эйлерова пути и цикла за $\mathcal{O}(V + E)$. Задачи: дополнение графа до Эйлерова, разбиение рёбер графа на минимальное число путей.
32. (a) dfs. De bruijn (строка, содержащая все строки длины n). Решение за $\mathcal{O}(\Sigma^n)$.
33. (b) dfs. 2-connectivity. Вершинная и рёберная. Выделение компонент.
34. (b) dfs. 2-SAT. Решение за $\mathcal{O}(n + m)$. Задачи: 2-List-Coloring.
35. (b) dfs. Матрица достижимости графа за $\mathcal{O}(VE)$, за $\mathcal{O}(\frac{VE}{w})$.

Правила сдачи

Вы записываетесь в [google-doc](#), получаете время сдачи, когда нужно прийти и сдать. В нужный момент вы приходите и говорите “я готов сдавать!”. Если есть свободный принимающий, в ответ вы получаете **три случайных числа** – номера вопросов в каждой из групп, и идёте **20 минут готовиться**. Если вы опоздали более чем на 5 минут, попадаете в живую очередь таких же опоздавших, у вас самый низкий приоритет, но как только кто-то осводится, вас тоже послушают.

При ответе билетов:

1. В **любых** спорных ситуациях подходим к лектору.
2. Лучше рассказать хоть что-то, чем ничего не сказать.
3. Вы готовились к экзамену! Вы должны помнить, что вы должны знать, что к какому билету относится.
4. Если вас спрашивают что-то не из билета, не из программы, не стесняйтесь говорить об этом.
5. Принимающий может в любой момент задать не сложный допвопрос из серии «что такое эйлеров путь?» или «за сколько вы умеете решать рюкзак?».
6. Если вы плохо знаете свой билет, можно вкратце проговорить, что вы всё-таки по теме помните, и заменить билет. Замена происходит так: принимающий мгновенно генерит псевдослучайное число отличное от предыдущего и говорит вам. Замена билета – минуса к оценке, зависящий от сложности билета.
7. Заменять билет можно только один раз.
8. Вы получаете 3 билета. По каждому у принимающего магическим образом складывается в голове оценка от 0 до 1 – уровень ваших знаний. Ваша оценка – сумма этих чисел, число от 0 до 3. Какие из этих чисел сообщать вам, а какие нет, решает принимающий. Вывод о ваших знаниях принимающий может описать вам не числом, а лишь на словах «тут вы почти ничего не знаете», «этот билет вы ответили идеально».

Алгоритм выставления оценки:

- a) Меньше 1.33 – двойка. Ваши знания не удовлетворительны.
- b) Больше 2.66 – пятёрка. Вы круты.
- c) 2.2 и больше – точно четвёрка.
- d) 1.8 и меньше – точно тройка.
- e) Если от 1.8 до 2.2, то идёт серия простых и не очень допвопросов, помогающих уточнить оценку.