

Вопросы к коллоквиуму по алгоритмам

СПБАУ, первый курс, осень, 2015/16 учебный год

База и структуры данных

1. (a) База. Асимптотика. Обозначения $o, \mathcal{O}, \Theta, \Omega, \omega$. Определение o через предел. Основные свойства.
2. (b) База. Преобразование сумм в интегралы. Пример “список делителей для всех чисел от 1 до n ”.
3. (b) База. Сравнение асимптотик: 5 терминов и теорема о сравнении асимптотик.
4. (b) База. Рекуррентные соотношения: доказательство по индукции, мастер-теорема (о простом рекуррентном соотношении). Версия без лишних логарифмов.
5. (b) База. Рекуррентные соотношения: теорема об экспоненциальном рекуррентном соотношении.
6. (a) База. Шесть способов увеличить константу времени работы. Кеш.
7. (a) DS. Массив фиксированного размера, списки, стек, очередь, дек.
8. (b) DS. Расширяющийся динамический массив. Время работы: real, average, amortized.
9. (a) DS. Очередь и дек на циклическом массиве. Стек с минимумом. Очередь с минимумом = два стека.
10. (b) DS. Очередь с минимумом, второй способ (дек минимумов).
11. (b) DS. Частичные суммы. Расширяющийся и сужающийся динамический массив. Время работы.
12. (a) DS. Бинарный поиск. 3 версии. Средства языка C++.
13. (a) DS. Хранение множеств и мультимножеств в виде отсортированных массивов. Средства языка C++.
14. (b) DS. Хеш-таблица на списках.
15. (b) DS. Хеш-таблица с открытой адресацией.
16. (b) DS. Сравнение двух реализаций хеш-таблиц. Средства языка C++.
17. (b) DS. Вектор: избавление от амортизации.
18. (c) DS. Очередь с минимумом: избавление от амортизации.
19. (c) DS. Хеш-таблица: избавление от амортизации.
20. (b) DS. Бинарная куча. HeapSort. Построение за линию. Обратные ссылки. Средства языка C++.
21. (c) DS. Аллокация памяти. Общий случай. Стек. Список. Оптимизация для C++.
22. (c) DS. Пополняемые структуры данных (Add \rightarrow Merge, Build \rightarrow Add). Пополняемый массив.
23. (b) DS. Разбор арифметического выражения. Унарные операции. Левая и правая ассоциативность.
24. (b) DS. Обобщённые два указателя на примере задачи “количество различных чисел на отрезке”.

Сортировки и кучи

25. (a) Sort. Квадратичные: Insertion, Selection, Bubble. Стабильность. IntroSort.
26. (b) Sort. Оценка снизу на число сравнений. Обобщения из практики.
27. (b) Sort. MergeSort. Нерекурсивная версия. Число инверсий.
28. (b) Sort. QuickSort без доказательства. Две версии partition: на python, двумя указателями. Способы Выбор медианы.
29. (b) Sort. QuickSort. Доказательство по индукции.
30. (b) Sort. QuickSort. Доказательство через дерево рекурсии и вероятность “сравнить два элемента”.
31. (a) Sort. Порядковая статистика за линейное время, рандомизированный алгоритм. Средства языка C++.
32. (b) Sort. Порядковая статистика за линейное время, детерминированный алгоритм.
33. (a) Sort. Сравнение сортировок за $\mathcal{O}(n \log n)$.
34. (b) Sort. Adaptive Heap Sort.
35. (b) Sort. Count Sort, Radix Sort за $\mathcal{O}(n \log_n m)$.
36. (c) Sort. Kirkpatrick Sort за $\mathcal{O}(n \log \log C)$.

37. (c) Sort. Bucket Sort. Две версии алгоритма. Три теоремы о времени работы.
38. (a) Inplace. Перечисление уже встречавшихся ранее. Unique. Rotate (2 способа). Merge за $\mathcal{O}(n \log n)$.
39. (b) Inplace. Merge за $\mathcal{O}(n)$. Средства языка C++.
40. (b) Heap. MinMax Heap. Подробная оценка времени работы.
41. (b) Heap. Leftist Heap, Skew Heap.
42. (b) Heap. Pairing Heap. Без доказательства времени работы.
43. (c) Heap. Van Emde Boas Tree. Добавление, извлечение минимума.
44. (a) Heap. Add за $\mathcal{O}(1)$, Min за $\mathcal{O}(1)$, Merge за $\mathcal{O}(1)$, DecreaseKey за $\mathcal{O}(1)$, ExtractMin за $\mathcal{O}(n)$. d -Heap.
45. (b) Heap. Binomial Heap. Оценка времени добавления $\mathcal{O}(1)$.
46. (c) Heap. Fibonacci Heap.
47. (b) Heap. Нижняя оценка на время построения бинарной кучи.
48. (b) Heap. Улучшение Add $\mathcal{O}(\log n) \rightarrow \mathcal{O}(1)$ и Merge $\mathcal{O}(\log n) \rightarrow \mathcal{O}(1)$ (bootstrapping).

Динамическое программирование

49. (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.
50. (a) DP. Задача про путь на матрице. Min путь, max путь, кол-во путей. Задача НОП. Восстановление ответа без дополнительных ссылок назад. Оптимизация памяти, если нет восстановления ответа.
51. (c) DP. Алгоритм Хиршберга для НОП за $\mathcal{O}(n^2)$ времени, $\mathcal{O}(n)$ памяти и восстановлением ответа.
52. (a) DP. Задача LCP. Задача “расстояние Левенштейна”.
53. (b) DP. Рюкзак. Решение за $\mathcal{O}(nW)$ времени, $\mathcal{O}(W)$ памяти и восстановлением ответа. Применение bitset. Проблемы с восстановлением ответа в случае стоимостей.
54. (b) DP. Задача LIS (НВП). Два решения за $\mathcal{O}(n^2)$. Решение за $\mathcal{O}(n \log n)$.
55. (a) DP. Подотрезки. Умножение матриц, расстановка скобок в выражении, удаление скобок.
56. (b) DP. Измельчение перехода. Задача про погрузку грузов на корабль, решения за $\mathcal{O}(n^4)$, $\mathcal{O}(n^3)$, $\mathcal{O}(n^2)$. Применение Хиршберга.
57. (b) DP. Возведение матрицы в степень: количество путей длины k , решение линейного рекуррентного соотношения.
58. (b) DP. Два указателя на примере задачи про выбор k точек на прямой среди n данных. Без док-ва.
59. (c) DP. Доказательство $p_{k-1, n} \leq p_{k, n} \leq p_{k, n+1}$
60. (b) DP. Трёхсторонний путь по матрице мин веса (про министерство). Оптимизация памяти до $\mathcal{O}(n\sqrt{m})$.
61. (a) DP. Операции с множествами, как с n -битовыми числами. Количество бит в числе за $\mathcal{O}(2^n)$. Сумма на подмножестве за $\mathcal{O}(2^n)$.
62. (a) DP. Subset Cover. Рюкзак на подмножествах.
63. (a) DP. Поиск старшего и младшего бита в числе.
64. (b) DP. Гамильтонов путь и цикл за $\mathcal{O}(2^n n)$.
65. (a) DP. Проверка независимости всех подмножеств вершин за $\mathcal{O}(2^n)$.
66. (b) DP. Перебор подмножеств и надмножеств. Покраска вершин графа в минимальное число цветов за $\mathcal{O}(3^n)$.
67. (c) DP. Оптимизация покраски вершин графа в минимальное число цветов с $\mathcal{O}(3^n)$ до $\mathcal{O}(2.44^n)$.
68. (b) DP. Динамика по профилю: количество красивых замощений за $\mathcal{O}(4^h w)$, $\mathcal{O}(8^h \log w)$.
69. (b) DP. Динамика по скошенному профилю: количество красивых замощений за $\mathcal{O}(2^h hw)$. Только рекурсивная реализация.

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 отвечать **без подготовки**. 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 спорных ситуациях подходим к лектору.
2. Лучше рассказать хоть что-то, чем ничего не сказать.
3. Вы готовились к экзамену! Вы должны хотя бы помнить, что вы должны знать, и что к какому билету относится. Если вас спрашивают что-то не из билета, не из программы, не стесняйтесь говорить об этом.
4. Принимающий может в любой момент задать не сложный допвопрос из серии “что такое дек?” или “за сколько вы умеете сортировать целые числа?”.
5. 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.
6. Заменять билет можно только один раз.
7. По каждой из тем 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. Сделанный вывод принимающий может описать не числом, а лишь на словах “тут вы почти ничего не знаете”, “а этот билет вы ответили идеально”.

Алгоритм выставления оценки:

- a) Меньше 1 – двойка. Ваши знания не удовлетворительны.
- b) Больше 2.5 – пятёрка. Вы круты.
- c) 2 и больше – точно четвёрка.
- d) 1.5 и меньше – точно тройка.
- e) Если от 1.5 до 2, то идёт серия простых и не очень допвопросов, помогающих уточнить оценку.