Вопросы к коллоквиуму по алгоритмам
СПб АУ, первый курс, осень, 2016/17 учебный год

Курсивом помечено то, что было разобрано на практике.

База и структуры данных

1. **(a)** База. Асимптотика. Обозначения $o, \mathcal{O}, \Theta, \Omega, \omega$. Определение o через предел. Основные свойства.
2. **(b)** База. Рекуррентные соотношения: алгоритм Карацубы.
3. **(b)** База. Рекуррентные соотношения: мастер-теорема (о простом рекуррентном соотношении).
4. **(c)** База. Рекуррентные соотношения: теорема об экспоненциональном рекуррентном соотношении.
5. **(a)** База. Числа Фибоначчи: вывод формулы. Доказательства рекуррентностей по индукции.
6. **(b)** База. Преобразование сумм в интегралы. Сумма гармонического ряда. Список делителей для всех чисел от 1 до n.
7. **(b)** База. Сравнение асимптотик: 5 терминов, теорема.
8. **(a)** База. Шесть способов увеличить константу времени работы. Примеры. Кеш. Быстрые операции.
9. **(a)** DS. Частичный суммы. Массив фикс. размера. Список двусвязный, односвязный, на массиве.
10. **(a)** DS. Вектор (расширяющийся массив). Стек, очередь, дек. *Средства языка `C++` и их сравнение.* Сравнение дека на списке и на массиве.
11. **(a)** DS. Очередь и дек на циклическом массиве. Стек с минимумом. Очередь с min через два стека.
12. **(b)** DS. *Очередь с минимумом, второй способ (дек минимумов).*
13. **(a)** DS. Разбор арифметического выражения со стеком за линейное время.
14. **(b)** DS. Время работы: real, average, amortized. Схема с монетками. Примеры.
15. **(b)** DS. Расширяющийся и сужающийся динамический массив. Доказательства времени работы.
16. **(a)** DS. Бинарный поиск. 3 версии. Средства языка `C++`.
17. **(b)** DS. Вещественный бинпоиск. *Корни многочлена $\forall$ степени. Определение границ бинпоиска.*
18. **(a)** DS. Хранение множеств и мультимножеств в виде сортированных массивов. Два указателя. Средства языка `C++`.

Структуры данных и кучи

19. **(a)** DS. Хеш-таблица на списках.
20. **(a)** DS. Хеш-таблица с открытой адресацией.
21. **(b)** DS. Сравнение двух реализаций хеш-таблиц. Средства языка `C++`.
22. **(b)** DS. Вектор: избавление от амортизации. 2 способа.
23. **(c)** DS. Очередь с минимумом: избавление от амортизации.
24. **(b)** DS. Хеш-таблица: избавление от амортизации.
25. **(b)** DS. Бинарная куча: хранение в массиве, Add, ExtractMin, DecreaseKey и обратные ссылки.
26. **(b)** DS. Бинарная куча: построение за линию, heap sort (inplace!). *Средства языка `C++`.*
27. **(b)** DS. Аллокация памяти. Стек. Список. Кучи с хеш-таблицей. Оптимизация для `C++`.
28. **(a)** DS. Преобразование операций. `Ничего` $\to$ `Del`, `Find` $\to$ `Del`, `Add` $\to$ `Merge`.
29. **(b)** DS. Пополняемые структуры данных: `Build` $\to$ `Add`, `Del`). Пополняемый массив.
30. **(b)** DS. Алгоритм Мо. Два указателя на примере задачи “количество различных чисел на отрезке”.
31. **(b)** Heap. Binomial Heap.
32. **(b)** Heap. Fibonacci Heap: Add и Merge за $\mathcal{O}(1)$, алгоритм для DecreaseKey.
33. **(c)** Heap. Fibonacci Heap: доказательство корректности, оценка времени работы.

Сортировки и кучи

34. **(a)** Sort. Квадратичные: Insertion, Selection, Bubble. Сравнение . Стабильность. IntroSort.
35. **(a)** Sort. Оценка снизу на число сравнений. *Обобщения из практики.*
36. **(b)** Sort. MergeSort. Нерекурсивная версия. *Число инверсий.*
37. **(b)** Sort. QuickSort без доказательства. Две версии partition: на python, двумя указателями. Способы выбора элемента для `partition`.
38. **(a)** Sort. Сравнение сортировок за $\mathcal{O}(n \log n)$.
39. **(b)** Sort. QuickSort. Доказательство по индукции с интегралами.
40. **(c)** Sort. QuickSort. Доказательство через дерево рекурсии и вероятность "сравнить два элемента".
41. **(b)** Sort. Порядковая статистика за линейное время, рандомизированный алгоритм. Средства `C++`.
42. **(b)** Sort. Порядковая статистика за линейное время, детерминированный алгоритм.
43. **(b)** Sort. Count Sort для чисел и для пар чисел, Radix Sort за $\mathcal{O}(n \log_n m)$.
44. **(b)** Sort. Bucket Sort. Две версии алгоритма. Три теоремы о времени работы.
45. **(c)** Sort. Kirkpatrick Sort за $\mathcal{O}(n \log \log C)$
46. **(c)** Heap. Van Embde Boas trees. Добавление элемента и извлечение минимума.
47. **(b)** *Inplace алгоритмы. Unique. Rotate. Merge за* $\mathcal{O}(n \log n)$.
48. **(b)** Heap. MinMax Heap. Подробная оценка времени работы.
49. **(b)** Heap. Leftist Heap, Skew Heap.
50. **(a)** Heap. *d-куча.* Куча, умеющая {Add, Min, Merge, DecreaseKey} за $\mathcal{O}(1)$, ExtractMin за $\mathcal{O}(n)$.
51. **(c)** Heap. Нижняя оценка на время построения бинарной кучи.

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 освободится, 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 дек?" или "чем MergeSort лучше QuickSort?".
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.