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

Сложность и рандомизированные алгоритмы

1. **(a)** Алгоритмически не разрешимая задача. Разрешимые: decision, search.
   Язык. Классы: `DTime`, `P`, `EXP`. Полиномиальная иерархия. `P` $\neq$ `EXP`.
2. **(a)** Классы `NP`, `coNP`. Примеры: `k-CLIQUE`, `MAX-CLIQUE`, `BH`, `HAM-PATH`, `PRIME`, `coPRIME`, `IS-SORTED`.
3. **(b)** Классы `NP`-hard, `NP`-complete, полиномиальное сведение. `BH` $\in$ `NP`-complete.
4. **(b)** Сведения `BH` $\to$ `CIRCUIT-SAT` $\to$ `SAT` $\to$ `3-SAT`.
5. **(a)** Сведения `3-SAT` $\to$ `k-IND` $\to$ `k-CLIQUE` $\to$ `VERTEX-COVER`.
6. Сведения L $\to$ SAT и применение SAT-solver
7. **(a)** Сведения search $\to$ decision. Search 3-SAT $\to$ search k-IND. Решение `NP`-hard задач на практике.
8. **(a)** 3 гипотезы.
9. **(a)** Классы: `RP`, `coRP`, `ZPP`. Сравнение с `NP`. Понижение ошибки.
10. **(a)** `ZPP` $=$ `RP` $\cap$ `coRP`
11. **(a)** Задачи: поиск квадратичного невычета, проверка на простоту (Ферма, Миллер-Рабин), `3-LIST-COLORING`, matrix multiplication testing.
12. **(b)** `BPP` и понижение ошибки.
13. **(a)** Применение случайных чисел: кодирование, подсчёт суммы чисел без разглашения. Парадокс дней рождений с доказательством.
14. **(a)** ρ-эвристика Полларда, факторизация чисел.
15. **(a)** Решения 3-SAT: детерминированный алгоритм $3^{n/2}$, его рандомизированная версия, random walk.
16. **(a)** Лемма Шварца-Зиппеля (формулировка). Матрица Татта и поиск совершенного паросочетания.

Кратчайшие пути

17. **(a)** bfs. Кратчайший путь за $\mathcal{O}(V+E)$. Сравнение с dfs.
    Матрица расстояний за $\mathcal{O}(VE)$. Граф кратчайших путей.
18. **(b)** bfs. 0-1-bfs. Модификация рёбер 1-k-графа. Целочисленный 1-k-bfs. Вещественный 1-k-bfs и A-B-bfs. Вещественный 0-1-bfs.s
19. **(b)** bfs. Версия за $\mathcal{O}(\frac{V^2}{w})$. Версия для 1-k-графа за $\mathcal{O}(E\log k)$.
20. **(a)** Дейкстра. Кратчайший путь за $\mathcal{O}(V^2)$, $\mathcal{O}(E\log V)$, $\mathcal{O}(E+V\log V)$.
21. **(a)** A^*. Доказательство корректности для графов с неравенством треугольника. Сравнение скорости с Дейкстрой. Пример, на котором A^* даёт преимущество.
22. **(a)** Флойд. Решение за $\mathcal{O}(V^3)$. Восстановление пути. Транзитивное замыкание за $\mathcal{O}(\frac{V^3}{w})$.
23. **(a)** Форд-Беллман. Решение за $\mathcal{O}(VE)$. Реализация на одномерном массиве. Восстановление пути.
24. **(b)** Форд-Беллман. Оптимизации: `break`, `queue`, `random_shuffle`.
25. **(b)** Джонсон. Идея потенциалов. Матрица расстояний за $\mathcal{O}(VE+V^2\log V)$.
26. **(c)** Гольдберг. Основная идея. Лечение одной вершины за $\mathcal{O}(V+E)$. $\log N$ фаз.
27. **(c)** Гольдберг. Лечение пачки из хотя бы $\sqrt{k}$ вершин за $\mathcal{O}(V+E)$.
28. **(b)** Отрицательный циклы. С помощью Флойда. С помощью Форд-Беллмана.
    Восстановление ответа, доказательство корректности.
29. **(a)** Min mean cycle за $\mathcal{O}(VE\log nC)$.
30. **(b)** Карп. Min mean cycle за $\mathcal{O}(VE)$.

DSU, MST, Жадность

31. **(a)** DSU на списках. Задача: для каждой пары вершин найти путь с минимальным max ребром.
32. **(b)** DSU на деревьях: эвристика сжатия путей, две версии ранговой эвристики.
    Доказательство того, что любая одна из трёх эвристик даёт время работы $\mathcal{O}(\log n)$.
33. **(b)** DSU на деревьях: доказательство $\mathcal{O}(m \log^* n)$.
34. **(c)** DSU на деревьях: доказательства $\mathcal{O}(m + n \log^* n)$, $\mathcal{O}(m \log^{**} n))$.
35. **(c)** DSU на деревьях: обратная функция Аккермана, доказательство $\mathcal{O}((n+m)A^{-1}(n))$.
36. **(a)** MST. Лемма о разрезе. Краскал. Прим. Доказательства.
37. **(a)** MST. Борувка. Время работы, доказательство.
38. **(a)** Approx. Salesman: 2-`OPT`, 1.5-`OPT`, $\nexists$ C-`OPT` алгоритма для графов без неравенства треугольника.
39. **(a)** Жадность. Гамильтонов путь. Правило Варнсдорффа. Переборное решение для произвольного графа.
40. **(a)** Жадность. Хаффман. Безпрефиксные коды. Поиск кодов, кодирование, декодирование. Поиск кодов по массиву частот за $\mathcal{O}(sort + |\Sigma|)$. Способы хранения дерева кодов.
41. **(a)** Жадность. Задачи на сортировку. Способы доказательства и вывод компаратора. 5 задач.
42. **(a)** Жадность. Задания с дедлайнами: выполнить все за $\mathcal{O}(n \log n)$, выполнить максимум за $\mathcal{O}(n^2)$.

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 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.