Вопросы к экзамену по алгоритмам
СПБ АУ, первый курс, осень, 2014/15 учебный год

1. Структуры данных. Offline, online, real time, amortized time. Вектор без амортизации. Хеш-таблица на списках без амортизации.
2. Аллокация памяти. Версия с осовобождением только "последнего выделенного" (стек). Версия для кусков памяти одинакового размера.
3. Преобразование структур данных `merge` → `add`; `add` → `merge`; `find` → `delete`; `build`, `get` → `merge`, `add`.
4. Сортировки: Adaptive Heap Sort, Bucket Sort. Сортировка n строк длины L.
   Преимущества Insertion Sort, Selection Sort.
5. Integer sorting: цифровая сортировка за $\mathcal{O}(n \log_n m)$; алгоритм Kirkpatrick'a за $\mathcal{O}(n \log \log m)$
6. **Stable** Inplace Merge за $[\mathcal{O}(n \log n), \mathcal{O}(\log n)]$, $[\mathcal{O}(n), \mathcal{O}(\min(|a|, |b|))]$. Inplace MinMax куча.
7. Нижняя оценка на build бинарной кучи. Нижняя оценка на скорость работы произвольной кучи.
8. k-heap, обратные ссылки для поиска элемента, Leftist Heap, Skew Heap, Bootstrapping.
9. Pairing heap. Доказательство амортизированного времени работы $\mathcal{O}(\sqrt{n})$.
10. Биномиальная куча. Куча Фибоначчи. DecreaseKey в куче Фибоначчи за $\mathcal{O}(1)$.
    Доказательство времени работы.
11. Задачи о рюкзаке. Выбор вещей максимального суммарного веса не более W. Выбор вещей максимальной суммарной стоимости и суммарного веса не более W. Решение обеих за $\mathcal{O}(nW)$ времени, $\mathcal{O}(W)$ памяти. Восстановление ответа для **первой** из двух задач (без стоимостей).
12. Решение задачи про погрузку ящиков на корабль за $\mathcal{O}(n^2)$ времени и $\mathcal{O}(n)$ памяти. Промежуточные и менее оптимальные решения.
13. Динамика по подмножествам: операции с множествами, подсчёт числа бит, суммы, страшего бита, младшего бита. Выделение всех независимых множеств за $\mathcal{O}(2^n)$. Покраска вершин графа за $\mathcal{O}(3^k)$.
14. Динамика по подмножествам: гамильтонов путь и цикл за $\mathcal{O}(2^n n)$.
15. Поиск мостов, точек сочленения, компонент рёберной и вершинной двухсвязности за линейное время.
16. Эйлеров путь, цикл в ориентированных и неориентированных графах. Поиск цикла Де Брюина на бинарных строках длины n за $\mathcal{O}(2^n)$.
17. Задача про оптимальную ориентацию графа. Решение за $\mathcal{O}(m^2)$.
18. Метод двух указателей: решение задачи про профессора и транзистры за $\mathcal{O}(n \log n)$; поиск количества различных чисел на отрезке за $\mathcal{O}((n+m)\sqrt{n})$.
19. Решение задачи: даны n точек на прямой, выбрать k так, чтобы сумма взвешенных расстояний до ближайшей выбранной была минимальна. Без доказательства.
20. Доказательство корректности решения задачи про выбор k точек ($p[n, k\text{-}1] \le p[n, k] \le p[n\text{+}1, k]$).
21. Модификации поиска в ширину: (0,1) граф; (1,k) граф; вещественные веса. Поиск кратчайшего пути за $\mathcal{O}(m + n\sqrt{k})$, $\mathcal{O}(m \log k)$.
22. Radix heap. Поиск кратчайшего пути за $\mathcal{O}(m + n \log k)$ с помощью radix heap.
23. Алгоритм Форда-Беллмана, реализация с $\mathcal{O}(n)$ памяти. Поиск отрицательного цикла, доказательство корректности.
24. Алгоритм Форда-Беллмана, оптимизации: **random-shuffle** (док-во улучшения в 1.5 раз), break, очередь, topsort, levit, SCC (stong-connectivity-components).
25. Алгоритм Борувки поиска MST за $\mathcal{O}(m \log \frac{n^2}{m})$.
26. Потенциалы, алгоритм Джонсона, алгоритм Гольдберга, версия за $\mathcal{O}(nm)$.
27. Оптимизация одной фазы алгоритма Гольдберга до $\mathcal{O}(m\sqrt{n})$. Поиск кратчайших путей в графе с целыми весами от $\text{-}N$ за $\mathcal{O}(m\sqrt{n} \log N)$.
28. Алгоритм Йена для поиска k-го кратчайшего пути за $\mathcal{O}(nk \cdot D)$ времени и $\mathcal{O}(nk)$ памяти.
29. Поиск цикла минимального среднего веса: бинарный поиск, Алгоритм Карпа за $\mathcal{O}(nm)$.
30. Алгоритм A^* для поиска кратчайшего пути в графах с неравенством треугольника. Доказательство оценки $\mathcal{O}(m \log n)$ на время работы. Примеры, на которых A^* лучше.