Зачёт по курсу “Алгоритмы обработки потоковых данных”

Казань, 2016

oparin.vsevolod [at] gmail [dot] com.

Таблица для зачета

Для зачета нужно набрать 15 баллов (из 17 возможных). Набирать баллы можно следующим образом.

- (1 балл) Решить один пункт одной задачи.
- (4 балла) Записать конспект лекции.
- (6 баллов) Написать визуализатор алгоритма.

Решение задачи. Решения по каждой задаче пишите подробно. Настолько подробно, чтобы ваш одногруппник мог без ваших подсказок его понять.

Все решения и вопросы присылайте на мой e-mail. В теме письма напишите “Казань. Поточные алгоритмы. 2016. <Имя Фамилия>”. Крайне желательно оформлять решения в LaTeX. Для простоты входа, я выкладываю исходник этого документа и шаблон на GitHub.

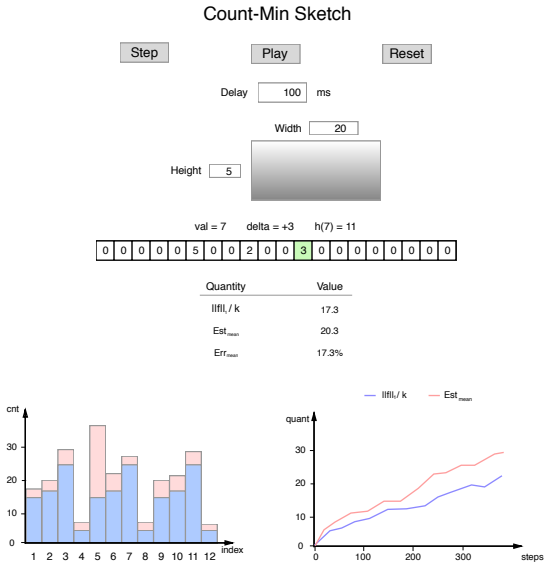
Конспект лекции. Перед тем, как писать конспект лекции, согласуйте это со мной. Я буду поддерживать в таблице с домашним заданием, кто какую лекцию взял. После того, как вы пришлете мне черновик, я могу попросить вас что-то исправить или добавить. Как только все исправления будут внесены, коспект засчитывается.

Конспект принимается только в LaTeX.

Визуализатор. Наиболее творческая часть зачета. Напишите визуализатор одного из алгоритмов. Я могу предложить вам написать такой для Count-Min скетча или Count скетча. Вы можете предложить свой. Для вдохновления посмотрите ссылки: раз, два, три.

Хочется, чтобы визуализатор был доступен широким массам. Я предлагаю его реализовать на JavaScript, потом я его куда-нибудь выложу и прикреплю ссылку к описанию курса. Если есть идеи как сделать лучше, буду рад услышать.

Я представляю визуализатор примерно так:



Теоретические задачи

Для простоты, считайте что длина потока m и размер множества n – величины одного порядка, т.е. $n = \Theta(m)$.

Задачи на последовательностях

1. Дана последовательность различных объектов $\sigma = \langle a_1, a_2, \cdots, a_m \rangle$, $a_i \in [n]$. Изначально длина последовательности неизвестна. Требуется за один проход выбрать случайно подмножество из k *различных* объектов, используя $O(k \log n)$ памяти. Гарантируется, что $k \geq n$.

2. Дана последовательность различных элементов $\sigma = \langle a_1, a_2, \cdots, a_m \rangle$, где $a_i \in [n]$. Требуется найти k-ую порядковую статистику: элемент последовательности, который встанет на k-ую позицию после сортировки. Разрешается выдать ответ с погрешностью $\varepsilon \cdot m$, т.е. вернуть такой элемент, чья позиция в отсортированной последовательности лежит в отрезке $[k - \varepsilon \cdot m, k + \varepsilon \cdot m]$. Решите задачу с вероятностью успеха $1 - \delta$, используя $O(\varepsilon^{-2} \cdot \log \delta^{-1} \log n)$.

   Что делать, если m заранее неизвестно?

3. Из набора чисел от 1 до n удалили все числа от l до r, оставшиеся перемешали. Найдите за один проход l и r. Память: $O(\log n)$.

4. Дана последовательность элементов $\sigma = \langle a_1, \cdots, a_m \rangle$, $a_i \in [n]$. В последовательности сушествует ровно два элемента, которые встречаются 1 раз. Остальные встречаются четное число раз.

   (a) Найдите оба элемента детерминированно за $O(\log^2 n)$.

   (b) Найдите оба элемента за $O(\log \delta^{-1} \log n)$ с вероятностью успеха $1 - \delta$.

   (c) Обобщите вероятностное решение для поиска k уникальных элементов за $O(k^2 \log n)$ с вероятностью успеха $\geq \frac{1}{2}$.

   (d) Оптимизируйте память до $O(k \log^2(k\delta^{-1}) \log n)$ при условии, что алгоритм работает с вероятностью успеха $\geq 1 - \delta$.

Задачи на графах

5. Дан связный ориентированный граф в виде последовательности ребер на n вершинах. Известно, что граф содержит Эйлеров путь, т.е. путь проходящий по всем ребрам ровно один раз. Эйлеров путь в ориентированном графе идет всегда по направлению ребра.

   (a) Найти начало и конец пути за один проход, детерминированно. Память $O(\log n)$.

   (b) Пусть из стартовой вершины пути исходит ровно одно ребро. Докажите, что для поиска второй вершины пути детерминированный алгоритм использует $\Omega(n)$ памяти.

6. Дан неориентированный граф на n вершинах, содержащий Эйлеров путь. Постройте алгоритм, который находит за один проход оба конца Эйлерова пути с вероятностью $1-\delta$. Память $O(\log \delta^{-1} \log n)$.

Нижние оценки

	$D^{\rightarrow}$	$R^{\rightarrow}_{\frac{1}{3}}$	D	$R_{\frac{1}{3}}$
INDEX	$\geq n$	$\geq n$	$\leq \lceil \log n \rceil$	$\leq \lceil \log n \rceil$
EQ	$\geq n$	$O(\log n)$	$\geq n$	$O(\log n)$
DISJ	$\Omega(n)$	$\Omega(n)$	$\Omega(n)$	$\Omega(n)$

7. Дана последовательность $\sigma = \langle a_1, \cdots, a_m \rangle$, $a_i \in [n]$. Каждый элемент $x \in [n]$ присутствует в последовательности не более одного раза. Докажите, что точный детерминированный алгоритм, находящий наименьшее число $x \notin \sigma$, использует $\Omega(n)$ памяти.

8. Дана последовательность уникальных чисел $\sigma = \langle a_1, \cdots, a_m \rangle$, $a_i \in [n]$. Инверсия последовательности – это пара индексов $i < j$ таких, что $a_i > a_j$. Четность последовательности – это четность числа инверсий. Докажите, что однопроходный детерминированный алгоритм, определяющий четность, использует $\Omega(n)$ памяти.

9. Докажите, что однопроходный алгоритм, который считает момент F_2 точно, использует $\Omega(n)$ памяти. Алгоритм может использовать случайные биты. Покажите, что если алгоритм использует p проходов, ему потребуется хотя бы $\Omega(n/p)$ памяти.

10. Покажите, что точный вероятностны однопроходный алгоритм для поиска медианы использует $\Omega(n)$ памяти.

11. Есть невзвешенный неориентированный граф G, данный в виде потока ребер. Граф построен на n вершинах. Требуется определить, есть ли в графе треугольник (три вершины, соединенные ребрами). Покажите, что однопроходный вероятностный алгоритм для этой задачи использует $\Omega(n^2)$ памяти.

Хэш-функции

12. Построим семейство хэш-функций $\mathcal{H} = \{h_i : \{0,1\}^n \to \{0,1\}^k\}_i$. Функции хэшируют битовые строки длины n в строки длины k следующим образом. Берем случайно равновероятно битовую матрицу $A \in \{0,1\}^{n \times k}$ и вектор $b \in \{0,1\}^k$. Определим хэш-функцию

$$h_{A,b}(x) = A \cdot x + b^T,$$

где все вычисления берутся по модулю 2. Докажите, что семейство хэш-функций 2-независимо.

13. Пусть есть семейство функций $\mathcal{H} = \{h_i : K \to V\}_i$. Каждая функция семейства инъективна, т.е. любых различных $k_1, k_2 \in K$ выполняется $h_i(k_1) \neq h_i(k_2)$. Пусть известно, что для любого множества $A \subseteq K$ и любого ключа $k \in A$ выполняется равенство:

$$\mathop{\mathbf{Pr}}_{h_i \leftarrow U(\mathcal{H})} \left[h(k) = \min_{a \in A} h(a) \right] = \frac{1}{|A|}.$$

Покажите, что для любых двух множеств $A, B \subseteq K$ выполняется условие:

$$\mathop{\mathbf{Pr}}_{h_i \leftarrow U(\mathcal{H})} \left[\min_{b \in B} h(b) = \min_{a \in A} h(a) \right] = \frac{|A \cap B|}{|A \cup B|}.$$