Практика по алгоритмам #4

Содержание

1. Задачи

1. Даны два массива a и b длины n, сгенерировать все попарные суммы $a_i + b_j$ в сортированном порядке.
   a) За $\mathcal{O}(n^2 \log n)$.
   b) За $\mathcal{O}(n^3)$ с использованием $\mathcal{O}(n)$ дополнительной памяти.
   c) За $\mathcal{O}(n^2 \log n)$ с использованием $\mathcal{O}(n)$ дополнительной памяти.
   d) За $\mathcal{O}(n^3)$ с использованием $\mathcal{O}(1)$ дополнительной памяти.
2. Дан массив из $n+1$ целого числа от 1 до n. Массив доступен только на чтение, есть $\mathcal{O}(1)$ дополнительной памяти. Найти за $\mathcal{O}(n)$ любое число, которое встречается хотя бы два раза.
3. Есть много разных сортировок. Есть даже такие, что работают дольше, чем квадрат. Оцените время работы в среднем следующих сортировок:
   a) `i = rand(), j = rand(), if a[min(i,j)] > a[max(i,j)] swap`
   b) `i = rand(), j = rand(), swap`
4. В свободное время Анка-пулеметчица любит сортировать патроны по серийным номерам. Вот и сейчас она только разложила патроны на столе в строго отсортированном порядке. Но тут Иван Васильевич распахнул дверь с такой силой, что все патроны на столе подпрыгнули и немного перемешались. Оставив ценные указания, Иван Васильевич отправился восвояси. Как оказалось, патроны перемешались не сильно. Каждый патрон отклонился от своей позиции не более чем на k. Всего патронов n. Помогите Анке отсортировать патроны.
   a) Отсортируйте патроны за $\mathcal{O}(nk)$.
   b) Отсортируйте патроны за $\mathcal{O}(n + I)$, где I — число инверсий.
   c) Докажите нижнюю оценку на время сортировки $\Omega(n \log k)$.
   d) Отсортируйте патроны за $\mathcal{O}(n \log k)$.
5. Дано n точек на плоскости. Соединить их
   a) $(n-1)$–звенной ломаной без самопересечений (не замкнутой);
   b) n–звенной ломаной без самопересечений (замкнутой)
   Оба пункта за $\mathcal{O}(n \log n)$.
6. Даны два массива a и b одинаковой длины.
   Нужно найти такую перестановку p, что
   a) $\sum_{i=1}^{n} a_{p_i} b_i \to \max$.
   b) $\sum_{i=1}^{n} a_{p_i} b_i \to \min$.
7. Дан набор из N отрезков $[a_i, b_i]$. Числа a_i, b_i — вещественные.
   a) Найти такое вещественное x, что $|\{i \colon x \in [a_i, b_i]\}|$ максимально.
   b) Длину объединения отрезков.
   c) Для каждого k посчитать, сколько точек на прямой покрыто ровно k отрезками.
8. Даны два массива из положительных чисел a и b. $|a| = |b| = n$.
   Выбрать массив $p \colon k$ различных чисел от 1 до n так, чтобы:
   a) $\sum_{i=1}^{k} a_{p_i} b_{p_i} \to \max$.
   b) $\frac{\sum_{i=1}^{k} a_{p_i}}{\sum_{i=1}^{k} b_{p_i}} \to \max$.
   c) $(\sum_{i=1}^{k} a_{p_i})(\sum_{i=1}^{k} b_{p_i}) \to \max$.

2. Разобранное в классе

1. **Сгенерировать все попарные суммы $a_i + b_j$ в сортированном порядке.**
   a) За $\mathcal{O}(n^2 \log n)$: отсортируем массив длины $O(n^2)$.
   b) За $\mathcal{O}(n^2 \log n)$ с использованием $\mathcal{O}(n)$ дополнительной памяти: отсортируем A, для каждого b_j сделаем копию массива A: $A + b_j$. Храним heap указателей на первый не использованный элемент в каждом из массивов. n^2 раз делаем extract min и add(next).
   c) За $\mathcal{O}(n^3)$ с использованием $\mathcal{O}(1)$ дополнительной памяти: пусть x — последнее выведенное число. Методом двух указателей за $\mathcal{O}(n)$ найдем миниммально число большее x и сколько раз оно встречается.
2. **Дан массив p из $n+1$ целого числа от 1 до n...** $x_0 = n + 1, \quad x_{i+1} = p[x_i] \neq x_0$. Осталось найти предпериод последовательности. Смотри предыдущую практику.

3a. **Случайная сортировка.** Каждый swap уменьшает число инверсий. $T(n, I)$ — время до очередного swap-а. Вероятность p попасть в инверсию равна $\frac{I}{n(n-1)/2}$, где I — число инверсий. $T(n, I) = 1 + (1 - p)T(n) \Rightarrow T(n) = \frac{1}{p} = \frac{n(n-1)/2}{I}$.
$\sum\limits_{I=1}^{n(n-1)/2} T(n, I) = n(n-1)/2 \sum\limits_{I=1}^{n(n-1)/2} \frac{1}{I} = \Theta(n^2 \log n)$.

3b. Утверждение: на каждом шаге наша перестановка близка к случайной, значит, с вероятностью $\frac{1}{n!}$ отсортирована.

4. **Дано n точек на плоскости. Соединить их**
   a) Ломаной: sort по x, при равенстве по y.
   b) Замкнутой ломаной: вокруг самой левой точки sort по углу, при равенстве по расстоянию.
5. **Скалярное произведение** Отсортируем a по возрастанию. Если хотим максимум, отсортируем b также по возрастанию, иначе по убыванию. Доказательство — трансверсальное неравенство. Попробуем поменять местами два элемента: $AB + ab \geq Ab + Ba \Leftrightarrow A(B - b) + a(b - B) \geq 0 \Leftrightarrow (A - a)(B - b) \geq 0$.

3. Домашнее задание

3.1. Обязательная часть

1. Даны n отрезков на прямой. Выбрать за $\mathcal{O}(n\log n)$ максимальное по размеру подмножества так, чтобы каждая точка была покрыта не более k раз.
   a) (2) Придумать алгоритм за $\mathcal{O}(n\log n)$.
   b) (2) Доказать корректность алгоритма.
2. Рассмотрим бинарную кучу. Пусть $d(v)$ = расстояние вниз от вершины до ближайшего "отсутствия ребенка" -1. Куча называется левацкой, если $\forall v\colon d(l[v]) \geq d(r[v])$.
   a) (1) докажите, что $size(v) \geq 2^{d(v)}$
   b) (2) напишите быстрый `Merge` для данной кучи, дайте оценку времени работы. Один вызов `Merge` должен работать за $\mathcal{O}(\log n)$.
3. (2) Практика.8b. За $\mathcal{O}(n \log M)$.
   M нужно придумать самим. Подсказка: нужен бинарный поиск.

3.2. Дополнительная часть

1. (3) Дано $2 \cdot n - 1$ коробок с черными и белыми шарами. В i-ой коробке находится w_i белых и b_i черных шаров. Всего в коробках находится W белых и B черных шаров. Требуется выбрать n коробок, чтобы суммарное число белых шаров в них было не менее $\frac{W}{2}$, а черных не менее $\frac{B}{2}$. Решить за $\mathcal{O}(n \log n)$.
2. (3) Придумайте, как модифицировать обычную бинарную кучу так, чтобы `Add` работал за амортизированное $\mathcal{O}(1)$.
3. (4) Практика.8c. За $\mathcal{O}(Polynom(n))$. Подсказка: возьмите, например, Пашу Кунявского, попросите решить эту задачу. Проследите внимательно за реакцией.