

Алгоритмы и структуры данных поиска. Итоговая Контрольная

Минский ШАД. Весна

23 мая 2015 г.

1. Проверить, что данный (n, m) -граф является деревом за $\mathcal{O}(n)$.
2. Пусть существует алгоритм A для детерминированного нахождения медианы в массиве за $\mathcal{O}(n)$, где n — длина массива. Предложите алгоритм для детерминированного нахождения k -й порядковой статистики в массиве за $\mathcal{O}(n)$.
3. Дана строка S . Добавим в бор все суффиксы этой строки. Назовём интересными вершинами те, у которых количество сыновей не равно единице. Докажите, что количество интересных вершин в боре не более, чем $2|S| + 3$.
4. Мальчик Лёша решил написать код для нахождения в отсортированном по возрастанию массиве a из n чисел первую такую позицию i , что $a_i > \text{value}$, либо возвращал -1 , если такого числа в массиве нет. Для этого он написал следующий код.

```
1 int first_bigger(int* a, int n, int value) {  
2     if (a[n - 1] <= value) return -1;  
3     int l = 0, r = n - 1;  
4     while (l < r) {  
5         int center = (l + r + 1) / 2;  
6         if (a[center] <= value) l = center + 1;  
7         else r = center;  
8     }  
9     return l;  
10 }
```

Конечно же этот код содержит ошибку. Исправьте ровно один символ так, чтоб он стал решать поставленную задачу.

Бонус: найдите 3 способа решить данную задачу.

5. Рассмотрим алгоритм T сжатия бинарных строк (т.е. алгоритм, который инъективно переводит бинарную строку в бинарную строку). Для оценивания качества алгоритма на строке S введём величину $k_T(S)$, равную длине образа строки S при применении алгоритма T . Назовём алгоритм T хорошим, если \exists такая константа c , что \forall таких S , что $|S| > c$ выполняется, что $k_T(S) < |S|$. Докажите, что хороших алгоритмов сжатия не существует.
6. Для всех $i = \overline{1 \dots n}$ определить количество таких троек натуральных чисел (a, b, c) , что $a \cdot b \cdot c = i$ за $\mathcal{O}(n \log n)$. Например, для числа 6 таких способов ровно 9.
7. Дан взвешенный связный (n, n) -граф. Вам разрешается сделать предобработку этого графа за $\mathcal{O}(n)$. После этого вам нужно:
 - (а) По паре вершин s и t определить максимальный $s-t$ поток в данном графе (веса — пропускные способности рёбер). Отвечать нужно за $\mathcal{O}(\log n)$.
 - (б) Определить минимальный разрез данного графа за $\mathcal{O}(1)$.

8. Два кольца заданы координатами своих центров и двумя радиусами. Необходимо определить площадь пересечения двух колец.
9. В городе Б есть n горизонтальных улиц и m вертикальных. На пересечении каждой вертикальной и горизонтальной улицы есть дом. Других домов в городе Б нет, а значит можно ввести прямоугольную систему координат на домах. Т.е. будем говорить, что в точке (x, y) расположен дом, если $1 \leq x \leq n$, $1 \leq y \leq m$. Вас как ответственного сотрудника поисковой компании интересует, как высок уровень любви к котикам в разных районах этого города. Для измерения этого показателя вы выбрали специальные числа h и w . Районом будем называть прямоугольник с противоположными сторонами в (x, y) и $(x + h, y + w)$ такой, что $1 \leq x \leq x + h \leq n$ и $1 \leq y \leq y + w \leq m$. Про каждый домик известно, как много раз в день хозяева дома используют поисковую систему для просмотра видео с котиками. Показатель любви района вычисляется как медианное количество раз, которое жители этого района заходят посмотреть котиков. По заданными числам n , m , h и w , а также показателям любви в каждом доме, вычислите район, который больше всего любит котиков. Гарантируется, что $h \cdot w$ нечётно.

Например рассмотрим следующий пример: $n = 4$, $m = 5$, $w = 3$, $h = 1$, красным отмечен искомый район.

1	1	0	1	2
1	1	0	1	3
1	0	2	2	4
0	2	2	1	8

10. Предложить алгоритм, который по заданному массиву целых чисел получит следующую в лексикографическом порядке перестановку этих чисел за $\mathcal{O}(n)$ времени и $\mathcal{O}(1)$ памяти.
11. Пусть есть алгоритм A , который за полиномиальное время умеет вычислять в произвольном взвешенном графе самый дешёвый (по сумме весов рёбер) путь из вершины x в вершину y . Предложите полиномиальный алгоритм поиска гамильтоновой цепи в графе.
12. У сильного и независимого мужчины есть n заначек с пивасиком. В процессе своего становления мужчина протоптал m тропинок. Каждая тропинка характеризуется тремя числами (a, b, c) — это значит, что она соединяет заначку с номером a и заначку с номером b , причём путь по этой тропинке занимает ровно c минут. Ходить по тропинке можно в обе стороны. От каждой заначки можно добраться до каждой, возможно через другие.

В какой-то момент сильный и независимый мужчина понял, что так больше продолжаться не может. Начать он решил с малого, а именно уничтожить лишние тропинки. Он хочет оставить граф тропинок связным, то есть после убирания тропинок их должно остаться ровно $n - 1$.

У мужчины с его закадычными друзьями есть славная традиция. Каждый год 31-го декабря сильный и независимый мужчина начинает праздничный обход всех заготовленных заначек. Он понимает, что дойдя до заначки с номером i , будет просто невежливо не потратить ровно d_i секунд на академическое исследование данной заначки. Причём, конечно, проводить исследование необходимо каждый раз при приходе к заначке, даже если эта заначка уже была обойдена раньше.

Для того, чтобы не пропустить основное празднество, мужчина хочет составить план обхода заначек. План должен начинаться в какой-то заначке s (на выбор мужчины), обходить все заначки, а потом возвращаться к заначке s .

Помогите мужчине составить такой план, время исполнения которого минимально за $\mathcal{O}(m \log m)$.

13. Дана строка S . Вам разрешается произвести препроцесс над строкой за время $\mathcal{O}(n \log n)$, где $|S| = n$.

Строка T называется доминируемой, если существует символ, который встречается в этой строке m раз, причём $m > \frac{|T|}{2}$.

Необходимо за $\mathcal{O}(\log^2 n)$ отвечать на запрос, правда ли, что подстрока $S[l \dots r]$ является доминируемой.

Бонус: Препроцесс за $\mathcal{O}(n)$, ответ за $\mathcal{O}(\log n)$.

Задание	1	2	3	4	5	6	7	8	9	10	11	12	13	Сумма
Баллы	0	0	0	0	0	0	0	0	0	0	0	0	0	0