

**Примеры заданий практической части вступительного испытания по информатике
(по направлению подготовки 01.03.02 Прикладная математика и информатика)
для восстановления после отчисления и перевода из других организаций в МФТИ**

на 2 семестр

1.

Дан отсортированный массив различных целых чисел $A[0..n-1]$ и массив целых чисел $B[0..m-1]$. Для каждого элемента массива $B[i]$ найдите минимальный индекс элемента массива $A[k]$, ближайшего по значению к $B[i]$. Время работы поиска для каждого элемента $B[i]$: $O(\log(k))$.

$n \leq 110000, m \leq 1000$.

in	out
3 10 20 30 3 9 15 35	0 0 2
3 10 20 30 4 8 9 10 32	0 0 0 2

2.

На числовой прямой окрасили N отрезков. Известны координаты левого и правого концов каждого отрезка (L_i и R_i). Найти сумму длин частей числовой прямой, окрашенных ровно в один слой.

In	Out
3 1 4 7 8 2 5	3

3.

Дано число $N < 10^6$ и последовательность целых чисел из $[-2^{31}..2^{31}]$ длиной N .

Требуется построить бинарное дерево, заданное наивным порядком вставки.

Т.е., при добавлении очередного числа K в дерево с корнем $root$, если $root \rightarrow Key \leq K$, то узел K добавляется в правое поддерево $root$; иначе в левое поддерево $root$.

Выведите элементы в порядке level-order (по слоям, “в ширину”)

in	out
3 2 1 3	2 1 3
3	1 2 3

1 2 3	
3 3 1 2	3 1 2
4 3 1 4 2	3 1 4 2

4.

Дано число $N < 10^6$ и последовательность пар целых чисел из $[-2^{31}..2^{31}]$ длиной N . Построить декартово дерево из N узлов, характеризующихся парами чисел $\{X_i, Y_i\}$. Каждая пара чисел $\{X_i, Y_i\}$ определяет ключ X_i и приоритет Y_i в декартовом дереве.

Добавление узла в декартово дерево сделайте следующим образом:

При добавлении узла (x, y) выполняйте спуск по ключу до узла P с меньшим приоритетом. Затем разбейте найденное поддереве по ключу x так, чтобы в первом поддереве все ключи меньше x , а во втором больше или равны x . Получившиеся два дерева сделайте дочерними для нового узла (x, y) . Новый узел вставьте на место узла P .

Построить также наивное дерево поиска по ключам X_i . При добавлении очередного числа K в дерево с корнем $root$, если $root \rightarrow Key \leq K$, то узел K добавляется в правое поддереве $root$; иначе в левое поддереве $root$.

Вычислить разницу глубин наивного дерева поиска и декартового дерева. Разница может быть отрицательна.

in	out
10 5 11 18 8 25 7 50 12 30 30 15 15 20 10 22 5 40 20 45 9	2
10 38 19 37 5 47 15 35 0 12 3 0 42 31 37 21 45	2

30 26	
41 6	

5.

Для каждой вершины определите расстояние до самой удаленной от нее вершины. Время работы должно быть $O(n)$.

Формат входных данных:

В первой строке записано количество вершин $n \leq 10000$. Затем следует $n - 1$ строка, описывающая ребра дерева. Каждое ребро – это два различных целых числа – индексы вершин в диапазоне $[0, n-1]$. Индекс корня – 0. В каждом ребре родительской вершиной является та, чей номер меньше.

Формат выходных данных:

Выход должен содержать n строк. В i -ой строке выводится расстояние от i -ой вершины до самой удаленной от нее.

in	out
3	2
0 1	1
1 2	2

на 3, 4, 5, 6, 7 или 8 семестр

1.

На числовой прямой окрасили N отрезков. Известны координаты левого и правого концов каждого отрезка (L_i и R_i). Найти сумму длин частей числовой прямой, окрашенных ровно в один слой.

In	Out
3	3
1 4	
7 8	
2 5	

2.

Дано число $N < 10^6$ и последовательность пар целых чисел из $[-2^{31}..2^{31}]$ длиной N .

Построить декартово дерево из N узлов, характеризующихся парами чисел $\{X_i, Y_i\}$.

Каждая пара чисел $\{X_i, Y_i\}$ определяет ключ X_i и приоритет Y_i в декартовом дереве.

Добавление узла в декартово дерево сделайте следующим образом:

При добавлении узла (x, y) выполняйте спуск по ключу до узла P с меньшим приоритетом. Затем разбейте найденное поддереве по ключу x так, чтобы в первом поддереве все ключи меньше x , а во втором больше или равны x . Получившиеся два дерева сделайте дочерними для нового узла (x, y) .

Новый узел вставьте на место узла P .

Построить также наивное дерево поиска по ключам X_i . При добавлении очередного числа K в дерево с

корнем root, если $root \rightarrow Key \leq K$, то узел K добавляется в правое поддерево root; иначе в левое поддерево root.

Вычислить разницу глубин наивного дерева поиска и декартового дерева. Разница может быть отрицательна.

in	out
10 5 11 18 8 25 7 50 12 30 30 15 15 20 10 22 5 40 20 45 9	2
10 38 19 37 5 47 15 35 0 12 3 0 42 31 37 21 45 30 26 41 6	2

3.

Для каждой вершины определите расстояние до самой удаленной от нее вершины. Время работы должно быть $O(n)$.

Формат входных данных:

В первой строке записано количество вершин $n \leq 10000$. Затем следует $n - 1$ строка, описывающая ребра дерева. Каждое ребро – это два различных целых числа – индексы вершин в диапазоне $[0, n-1]$. Индекс корня – 0. В каждом ребре родительской вершиной является та, чей номер меньше.

Формат выходных данных:

Выход должен содержать n строк. В i -ой строке выводится расстояние от i -ой вершины до самой удаленной от нее.

in	out
3 0 1 1 2	2 1 2

4.

Требуется отыскать самый выгодный маршрут между городами. Требуемое время работы $O((N+M)\log N)$, где N -количество городов, M -известных дорог между ними.

Оптимизируйте ввод

Формат входных данных.

Первая строка содержит число N – количество городов.

Вторая строка содержит число M - количество дорог.

Каждая следующая строка содержит описание дороги (откуда, куда, время в пути).

Последняя строка содержит маршрут (откуда и куда нужно доехать).

Формат выходных данных.

Вывести длину самого выгодного маршрута.

in	out
6	9
9	
0 3 1	
0 4 2	
1 2 7	
1 3 2	
1 4 3	
1 5 3	
2 5 3	
3 4 4	
3 5 6	
0 2	

5.

Дан неориентированный связный граф. Требуется найти вес минимального остовного дерева в этом графе.

Вариант 1. С помощью алгоритма Прима.

Вариант 2. С помощью алгоритма Крускала.

Вариант 3. С помощью алгоритма Борувки.

Ваш номер варианта прописан в ведомости.

Формат входного файла.

Первая строка содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \leq n \leq 20000$, $0 \leq m \leq 100000$).

Следующие m строк содержат описание ребер по одному на строке.

Ребро номер i описывается тремя натуральными числами b_i , e_i и w_i — номера концов ребра и его вес соответственно ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 100000$).

Формат выходного файла.

Выведите единственное целое число - вес минимального остовного дерева.

in	out
----	-----

4 4 1 2 1 2 3 2 3 4 5 4 1 4	7
---	---

6.
Найдите все вхождения шаблона в строку. Длина шаблона – p, длина строки – n.
Время $O(n + p)$, доп. память – $O(p)$.

Формат входного файла

Шаблон, символ перевода строки, строка.

Формат выходного файла

Позиции вхождения шаблона в строке.

Время: 100мс, память 3Мб.ц

stdin	stdout
aba abacababa	0 4 6