

Graph [17]

А. Система непересекающихся множеств

1 секунда, 256 мегабайт

Реализуйте систему непересекающихся множеств, и в ответ на каждый запрос объединения двух множеств, выведите размер получившегося множества.

Входные данные

В первой строке вам дано два числа  $N$  и  $M$  ( $1 \leq N, M \leq 2 \cdot 10^5$ ) — количество элементов в системе и количество запросов. Далее следуют  $M$  строк с запросами в виде пар чисел  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq N$ ) — элементы, множества которых необходимо объединить.

Выходные данные

В ответ на каждый запрос выведите единственное число — количество элементов в полученном после слияния множестве.

входные данные
4 2 1 2 3 4
выходные данные
2 2

входные данные
4 5 1 2 1 3 3 2 1 4 4 3
выходные данные
2 3 3 4 4

В. Система непересекающихся множеств с данными

1 секунда, 256 мегабайт

Реализуйте систему непересекающихся множеств, и в ответ на каждый запрос объединения двух множеств, выведите сумму всех идентификаторов элементов в множестве, минимальный и максимальный идентификаторы.

Входные данные

В первой строке вам дано два числа  $N$  и  $M$  ( $1 \leq N, M \leq 2 \cdot 10^5$ ) — количество элементов в системе и количество запросов. Далее следуют  $M$  строк с запросами в виде пар чисел  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq N$ ) — элементы, множества которых необходимо объединить.

Выходные данные

В ответ на каждый запрос три числа — сумму всех идентификаторов элементов в полученном множестве, минимальный и максимальный идентификаторы в нём.

входные данные
4 2 1 2 3 4
выходные данные
3 1 2 7 3 4

входные данные
4 5 1 2 1 3 3 2 1 4 4 3
выходные данные
3 1 2 6 1 3 6 1 3 10 1 4 10 1 4

С. Минимальное остовное дерево

1 секунда, 256 мегабайт

Реализуйте поиск минимального остова в графе.

Входные данные

В первой строке вам дано два числа  $N$  и  $M$  ( $1 \leq N, M \leq 2 \cdot 10^5$ ) — количество вершин и рёбер в графе. Далее следуют  $M$  строк с описанием рёбер в виде троек чисел  $u_i, v_i$  и  $w_i$  ( $1 \leq u_i, v_i \leq N, 1 \leq w_i \leq 2 \cdot 10^5$ ) — пара вершин соединяемых ребром и вес ребра.

Выходные данные

Выведите единственное число — вес минимального остова.

входные данные
3 3 1 2 10 2 3 1 3 1 2
выходные данные
3

входные данные
4 5 1 2 5 2 3 2 3 4 3 4 1 1 1 3 4
выходные данные
6

Д. Разрушение графа

1 секунда, 256 мегабайт

Промоделируйте процесс разрушения графа, при котором последовательно разрушаются рёбра соединяющие пары вершин. Гарантируется что в конце процедуры все рёбра в графе уничтожены.

Входные данные

В первой строке вам дано два числа  $N$  и  $M$  ( $1 \leq N, M \leq 2 \cdot 10^5$ ) — количество вершин и рёбер в графе. Далее следуют  $M$  строк с описанием рёбер в порядке, в котором они разрушались, в виде пар чисел  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq N$ ) — пара вершин соединяемых ребром.

Выходные данные

После каждого запроса выведите количество компонент связности в оставшемся графе.

входные данные
3 3 1 2 2 3 3 1

выходные данные
1
2
3

входные данные
4 5
1 2
2 3
3 4
4 1
1 3
выходные данные
1
2
2
3
4

Е. Фэнтези

2 секунды, 64 мегабайта

Василий — писатель и, как ему кажется, очень неплохой. Поэтому он решил создать свою собственную фэнтезийную вселенную. Он знает, что для успеха фэнтезийной вселенной обязательно нужна подробная карта, а также он знает что для большего драматургического эффекта границы между владениями различных фракций должны проходить по живописным естественным границам типа гор и рек, чтобы там проводить наиболее эпичные бои.

Он уже набросал расположение естественных границ и расположил на карте столицы всех фракций, теперь он просит вас обозначить территории, которые будут принадлежать каждой фракции. Фракции должны принадлежать все области карты, в которые можно попасть из столицы переходя между клетками только по сторонам.

Входные данные

В первой строке даётся два числа  $N$  и  $M$  ( $1 \leq N, M \leq 1000$ ) — количество строк и столбцов в карте мира. В следующих  $N$  строках дана карта мира, символом '#' обозначаются естественные границы, '.' свободные области, столицы фракций обозначены маленькими латинскими буквами. Гарантируется, что между столицами двух фракций не существует пути по свободным областям.

Выходные данные

Выведите  $N$  строк по  $M$  симфолов в каждой, итоговую карту в которой для каждой фракции все области принадлежащие ей будут помечены буквой соответствующей столицы.

входные данные
3 3
a.#
.#.
#b.
выходные данные
aa#
a#b
#bb

входные данные
3 5
a#.#.
.#.#.
.#.#b

выходные данные
a#.#b
a#.#b
a#.#b

F. Иерархическая кластеризация

1 секунда, 256 мегабайт

Кластерный анализ — многомерная статистическая процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы. Задача кластеризации относится к статистической обработке, а также к широкому классу задач обучения без учителя.

При иерархическом подходе к кластеризации предполагается наличие вложенных групп (кластеров различного порядка), а алгоритмы в свою очередь подразделяются на агломеративные (объединительные) и дивизивные (разделяющие).

Вам дан набор точек на прямой, от вас требуется реализовать один из вариантов иерархического агломеративного алгоритма кластеризации.

На начальном этапе каждая точка является кластером из одного элемента. На каждом шаге алгоритма необходимо найти два кластера, расстояние между ближайшими точками которых минимально и объединить их в новый кластер. В случае неоднозначности выбора такой пары, выбирается самая левая. Такие шаги повторяются пока не останется один кластер содержащий в себе все точки.

На каждой итерации алгоритма выведите границы кластеров, которые были объединены на этой итерации. Сначала границы левого кластера, затем правого.

Входные данные

В первой строке входных данных вам дано одно число  $N$  ( $1 \leq N \leq 10^5$ ) — количество точек в наборе. В следующей строке вам дано  $N$  точек  $p_i$  ( $|p_i| \leq 10^8$ ), которые необходимо кластеризовать, все точки попарно различны.

Выходные данные

На каждой итерации алгоритма выведите границы двух отрезков которые были объединены.

входные данные
5
10 -10 6 -6 0
выходные данные
-10 -10 -6 -6
6 6 10 10
-10 -6 0 0
-10 0 6 10

входные данные
5
10000 11000 11100 11110 11111
выходные данные
11110 11110 11111 11111
11100 11100 11110 11111
11000 11000 11100 11111
10000 10000 11000 11111