1) Сортировка, жадный алгоритм

2) Бинпоиск по ответу

3) Линейная динамика

4) Двумерная динамика

5) BFS или DFS

6) Кратчайший путь + бин поиск по ответу

7) Остовы + DSU + полиномиальное хеширование

8) Дерево отрезков суммы

1. Дед мороз приехал на праздник к ребятам в детский сад. У Деда мороза с собой мешок с подарками, каждый подарок характеризуется ценностью, натуральным значением *P*. Каждый ребенок хочет получить какой-нибудь подарок и будет доволен, если ценность подарка будет не ниже некоторого натурального значения *С*. Помогите Деду морозу так распределить подарки между ребятами, чтобы количество довольных ребят было максимально. Каждый ребенок может получить не более одного подарка.

**Входные данные**

В первой строке два числа N, M количество детей и количество подарков соответственно(0< N,M <105). В следующей строке вводятся значения, пожелания детей *C*(0< Ci <1000). Далее вводятся значения ценностей подарков *P*(0<Pi<1000).

**Выходные данные**

Программа выводит одно число – количество довольных детей.

Примеры

**Входные данные**

3 2

1 2 3

1 1

**Выходные данные**

1

**Входные данные**

4 3

4 2 3 5

10 1 1

**Выходные данные**

1

1. Логистической компании необходимо доставить грузы в пункт назначения. Из-за большого количества заказов компания может выделить на доставку в этот пункт назначения только один грузовик на все дни доставки, грузовик может совершать только один рейс в день в пункт назначения и обратно, у компании есть возможность выбрать любую грузоподъемность грузовика. Для заказчика важно, чтобы очередность доставки грузов не менялась, поэтому нельзя менять порядок грузов. Определите минимальную грузоподъемность грузовика C, при которой все грузы можно перевезти не более чем за d дней.

**Входные данные**

В первой строке вводится натуральные числа d количество дней и N количество грузов (0< N <105). В следующей строке вводятся веса грузов w(0<wi<2\*109)

**Выходные данные**

Программа выводит одно число – минимальную грузоподъемность грузовика.

Примеры

**Входные данные**

5 10

1 2 3 4 5 6 7 8 9 10

**Выходные данные**

15

**Входные данные**

3 6

3 2 2 4 1 4

**Выходные данные**

6

1. Лягушка прыгает по лестнице снизу вверх. На старте у лягушки есть некоторый запас энергии. Посещение каждой ступеньки, включая стартовое положение лягушки, требует платы в виде расхода некоторого количества энергии. Также каждая ступенька обладает свойством, задающим максимальный шаг прыжка, на который лягушка может прыгнуть вверх с этой ступеньки. Определите минимальный запас энергии лягушки необходимый для достижения последней ступеньки.

**Входные данные**

В первой строке вводится натуральное число N количество ступенек (0< N <105), нумерация ступенек начинается с 0(стартовая позиция лягушки). Во второй строке вводятся стоимости расхода энергии за посещение i-ой ступеньки C(0<Ci<2\*109). В следующей строке для каждой ступеньки вводятся значения длины максимального прыжка с текущей ступеньки H(0< Hi <105), например, Hi = 4 означает, что с i-ой ступеньки лягушка может прыгнуть на 1, 2, 3 или 4 ступеньки вверх.

**Выходные данные**

Программа выводит одно число – начальный минимальный запас энергии лягушки.

Примеры

**Входные данные**

4

1 2 3 4 1

2 2 3 1 0

**Выходные данные**

5

**Входные данные**

3

10 20 30 40

1 1 1 0

**Выходные данные**

100

1. Археолог исследует древние руины. У археолога есть карта руин, которую можно представить как матрицу, в которой каждая клетка представляет собой какую-то комнату руин. При исследовании необходимо руководствоваться следующими правилами: перемещаться между комнатами по карте можно вправо, вниз, по диагонали вправо-вниз, при этом есть ограничение на количество переходов между комнатами, и есть комнаты, которые нельзя посетить. В комнатах есть ценные артефакты и ловушки, ценность артефакта обозначается целым положительным числом, ловушки обозначаются целым отрицательных числом, стены обозначаются символом “x”. Археолог начинает исследование из левого верхнего угла карты, выход из руин находится в правом нижнем углу карты. Помогите археологу выйти из руин и собрать максимальную сумму на выходе из руин.

**Входные данные**

В первой строке вводится натуральные числа N и M, размеры руин (количество комнат) на карте (0< N, M <105).   
Далее вводится карта руин, в каждой ячейке может быть записано одно из трёх значений: положительное число (0<Pij<105), отрицательное число (-105<Pij<0), значение “x”. В следующей строке вводится количество шагов H(0< H <= 2 \* 105) Шагом считается один переход из одной клетки в другую.

**Выходные данные**

Программа выводит одно число – максимальную сумму, которую может получить археолог, если получить сумму невозможно (при заданных ограничениях нельзя выйти из руин или сумма является отрицательным числом), вывести «NO».

Примеры

**Входные данные**

3 3

1 x 3

4 x 6

7 x 9

5

**Выходные данные**

NO

**Входные данные**

3 3

1 x 3

4 5 6

7 8 9

4

**Выходные данные**

30

**Входные данные**

3 3

1 x 3

4 5 6

-100 -100 9

4

**Выходные данные**

25

1. Вася попал в заброшенный старый дом. В доме большое количество комнат. Комната может содержать набор ключей, каждый из которых открывает определенную другую комнату. Некоторые комнаты могут быть заблокированы и попасть в них можно, подобрав соответствующий ключ в другой комнате. Помогите Васе определить, можно ли посетить все комнаты, если этого сделать нельзя выведите в порядке возрастания номера комнат, которые посетить нельзя. Изначально попав в дом, Вася оказался в комнате с номером 0.

**Входные данные**

В первой строке вводится натуральное число N количество комнат (0< N <105), нумерация комнат начинается с 0. Далее в каждой строке вводится номер комнаты и список ключей в этой комнате

**Выходные данные**

Программа выводит YES, если можно посетить все комнаты, в противном случае NO, если какие-то комнаты посетить нельзя, и в следующий строке в порядке возрастания номера этих комнат.

Примеры

**Входные данные**

3

0 1

1 2

**Выходные данные**

YES

**Входные данные**

4

0 1 2

1 2

**Выходные данные**

NO

3

**Входные данные**

4

1 2 3

**Выходные данные**

NO

1 2 3

1. Группа альпинистов планирует поход в горы. На горе есть места для отдыха, базы. Между базами есть горные тропинки, каждая тропинка имеет определенное время, которое необходимо потратить, чтобы перейти из одной базы в другую.   
   Горы дело серьезное и опасное, поэтому для каждой тропинки есть категория сложности, группа может пройти по тропе только, если все альпинисты в группе имеют подготовку не меньше, чем категория сложности горной тропы. Уровень подготовки – значение общее для всей группы. Определите минимальное время, за которое группа может подняться от подножия горы (база 0) до вершины, при том, что уровень подготовки для группы должен быть минимальным. При этом остановки на базе не учитываются во времени подъема

**Входные данные**

В первой строке вводится натуральные числа N и M количество баз и троп (0< N, M <105), нумерация баз начинается с 0, N – 1 база располагается на вершине. Далее в каждой строке записываются номера баз, связанные тропами, время прохода по тропе T(0<T<105) и уровень сложности(0<P<105). По тропе можно перемещаться в обоих направлениях.

**Выходные данные**

Программа выводит минимальное время подъёма на N-1 вершину и минимальное значение уровня подготовки группы. Если добраться до нужной вершины нельзя программа выводит -1

Примеры

**Входные данные**

3 2

0 1 5 10

1 2 3 20

**Выходные данные**

8 20

**Входные данные**

4 4

0 1 10 50

0 2 5 100

1 3 1 70

2 3 8 60

**Выходные данные**

11 70

**Входные данные**

4 4

0 1 5 10

0 2 10 5

1 3 7 15

2 3 3 8

**Выходные данные**

13 10

1. Команда исследователей в заброшенных руинах нашла карту сокровищ. Сокровище спрятано где-то на заброшенном архипелаге. Чтобы получить сокровище необходимо активировать древний механизм. Для активации необходимо соединить между собой острова архипелага мостами, так чтобы можно было из любого острова попасть на любой другой остров. На каждом острове установлена стела с уникальными древними символами. После построения мостов необходимо вычислить магический ключ. Магический ключ вычисляется по следующему алгоритму: все надписи на стелах сортируются в лексикографическом порядке, затем объединяются в одну надпись, вычисляется полиномиальный хеш получившейся строки (основание 31, по модулю 109+7). Бюджет команды исследователей ограничен, поэтому суммарная стоимость построения мостов должны быть минимальной. Помогите исследователям уложится в бюджет при постройке мостов и вычислить магический ключ активации древнего механизма, чтобы получит сокровище.

**Входные данные**

В первой строке вводится натуральные числа N и M количество островов архипелага и мостов соответственно (0< N, M <105). В следующих N строках вводятся N надписей, сначала для первого острова, потом для второго и так далее, нумерация островов считается с 1. В следующих M строках вводятся возможные варианты мостов. В каждой такой строке вводятся номера двух островов и стоимость постройки моста С (0< С <105)

**Выходные данные**

Программа выводит минимальную стоимость соединения островов мостами. В следующей строке магический ключ активации. В случае, если соединить мостами нельзя программа выводит -1.

Примеры

**Входные данные**

3 2

alpha

beta

gamma

1 2 5

2 3 10

**Выходные данные**

15

18034296

**Входные данные**

2 1

fire

water

1 2 8

**Выходные данные**

8

40775623

**Входные данные**

5 7

alpha

beta

gamma

delta

epsilon

1 2 3

2 3 4

3 4 5

4 5 6

1 3 2

2 4 7

1 5 8

**Выходные данные**

14

92592944

1. В дендропарке выращивают редкие виды деревьев. Садовник ведёт статистику роста деревьев, все деревья имеют определенную высоту и некоторую скорость роста, значение, на сколько условных единиц каждое дерево увеличивается в день. Все деревья пронумерованы числами от 1 до N. Деревьев очень много, помогите садовнику написать программу, которая будет по некоторому запросу номеров деревьев из диапазона выдавать суммарную высоту деревьев в этом диапазоне. Программа должна учитывать изменения роста

**Входные данные**

В первой строке вводится натуральные числа N и Q количество деревьев и запросов соответственно (0< N, Q <105). В следующей строке вводятся начальные высоты деревьев H(0<= H <=105). В третьей строке вводятся скорости роста деревьев. Гарантируется, что в процессе роста дерева его высота не превысит значения 2\*109. Далее идут Q запросов двух видов:

* 1. s a b – вычисляется суммарный рост деревьев c номерами в диапазоне от a до b включительно (0 < a <= b <= N)
  2. grow – прошел день, все деревья выросли на соответствующие им значения скорости роста

**Выходные данные**

Для каждого запроса первого вида программа должна выдавать суммарную высоту деревьев

Примеры

**Входные данные**

3 3

10 20 30

1 2 3

s 1 3

grow

s 1 3

**Выходные данные**

60

66

**Входные данные**

2 2

5 10

1 1

grow

s 1 2

**Выходные данные**

17

**Входные данные**

4 3

1 2 3 4

0 0 0 0

s 1 1

s 4 4

s 1 4

**Выходные данные**

1

4

10