

Яндекс. Тренировки по алгоритмам 2.0, занятие 6 (А)

D. Эльфы и олени

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Скоро новый год и Санта-Клаус уже начал готовить свою волшебную оленью упряжку, на которой он развозит подарки детям. Известно, что упряжку везут несколько волшебных оленей, на каждом из которых едут два эльфа.

Но волшебные олени – строптивые животные, поэтому не любые два эльфа могут ехать на любом олене. А именно, каждый олень характеризуется некоторой строптивостью a_i , а каждый эльф – темпераментом b_j . Два эльфа j и k могут ехать на i -м олене в том и только в том случае, если либо $b_j < a_i < b_k$, либо $b_k < a_i < b_j$.

Чтобы его появление было максимально зрелищным, Санта-Клаус хочет, чтобы в его упряжке было как можно больше оленей. Про каждого оленя Санта знает его строптивость, а про каждого эльфа – его темперамент.

Помогите Санте выяснить, какое максимальное количество оленей он сможет включить в упряжку, каких оленей ему следует выбрать, и какие эльфы должны на них ехать.

Формат ввода

В первой строке вводятся два целых числа m и n – количество оленей и эльфов, соответственно ($1 \leq m, n \leq 100\,000$).

Вторая строка содержит m целых чисел a_i – строптивость оленей ($0 \leq a_i \leq 10^9$). В третьей строке записаны n целых чисел b_i – темперамент эльфов ($0 \leq b_i \leq 10^9$).

Формат вывода

В первой строке выведите одно число k – максимальное количество оленей, которое Санта-Клаус может включить в свою упряжку. В следующих k строках выведите по три целых числа: $d_i, e_{i, 1}, e_{i, 2}$ – для каждого оленя в упряжке выведите его номер и номера эльфов, которые на нем поедут. Если решений несколько, выведите любое.

И эльфы, и олени пронумерованы, начиная с единицы, в том порядке, в котором они заданы во входных данных.

Пример

Ввод

4 6
2 3 4 5
1 3 2 2 5 2

Вывод

2
1 1 2
2 4 5

```

1 def groups(count, deers, elves):
2     """
3     Функция моделирование групп
4     :param count: количество используемых пар эльфов
5     :param deers: список оленей
6     :param elves: список эльфов
7     :return: bool - получилось ли создать группу по используемым параметрам; группы эльфов и оленей
8     """
9     cur_deer = 0 # индекс текущего оленя
10    answer = [] # массив групп
11    # проходим по левым эльфам в группах
12    for lf_elf in range(count):
13        rg_elf = len(elves) - count + lf_elf # индекс правого эльфа в группе
14        # пока олени левее левого эльфа - пропускаем
15        while cur_deer < len(deers) and deers[cur_deer][0] <= elves[lf_elf][0]:
16            cur_deer += 1
17        # если оленей не осталось или очередной правее правого эльфа - ответ False
18        if cur_deer == len(deers) or deers[cur_deer][0] >= elves[rg_elf][0]:
19            return False, []
20        # добавляем к ответу группу (олень, эльфы)
21        answer.append((deers[cur_deer][1], elves[lf_elf][1], elves[rg_elf][1]))
22        cur_deer += 1 # следующий олень
23    # возвращаем флаг и ответ
24    return True, answer
25
26
27 m, n = map(int, input().split()) # количество оленей и эльфов
28 deers = [(deer, i + 1) for i, deer in enumerate(list(map(int, input().split())))] # массив пронумерованных оленей
29 elves = [(elf, i + 1) for i, elf in enumerate(list(map(int, input().split())))] # массив пронумерованных эльфов
30 # сортируем массивы
31 deers.sort()
32 elves.sort()
33
34 # правый бин поиск
35 lf, rg = 0, min(m, n // 2) # левый, правый указатели
36 while lf < rg:
37     mid = (lf + rg + 1) // 2 # количество используемых пар
38

```

Отправить

Предыдущая

Следующая