

**Межрегиональные предметные олимпиады КФУ**  
**профиль «Информатика»**  
**заключительный этап**  
**2020-2021 учебный год**  
**10 класс**

Для каждой задачи необходимо написать программу на одном из предложенных языков программирования: C, C++, Python, Java, C#, Pascal, алгоритмический язык.

Для каждой задачи временное ограничение: 2 секунды (время работы программы).

Для каждой задачи ограничение по памяти: 256 Мб.

**Задача № 1 (20 баллов)**

На заводе по производству наборов сладостей наборы составляются следующим образом: изначально в набор кладут  $k$  сладостей, цены которых  $a_1, \dots, a_k$ , затем, чтобы уравнять получившийся набор с другими  $N$  наборами на складе, этот набор сравнивают со всеми наборами на складе в порядке нумерации (наборы пронумерованы от 1 до  $N$ ) и делают следующую операцию: если самая дорогая сладость в новом наборе дороже самой дорогой из  $i$ -го набора со склада, то надо самую дорогую из нового набора поменять на самую дешевую из  $i$ -го набора со склада. Какова будет суммарная стоимость набора после такого “уравнивания”?

**Входные данные**

В первой строке дано  $k$  – количество сладостей в новом наборе, затем  $k$  чисел через пробел  $a_1, \dots, a_k$  – стоимости сладостей в наборе. В третьей строке дается  $N$  – число подарков на складе, затем  $N$  строк по 2 числа в строке – стоимость самой дорогой и самой дешевой сладости в  $i$ -том наборе на складе. Все заданные числа положительные целые.

$$1 \leq N \leq 1000,$$

$$1 \leq k \leq 1000,$$

$$1 \leq a_i \leq 10^9$$

**Выходные данные**

Итоговая стоимость набора.

**Пример**

INPUT	OUTPUT
5	12
3 4 2 10 3	
3	
11 1	
3 2	
3 2	

**Задача № 2 (20 баллов)**

В отеле поселились  $n$  групп туристов, которые хотят поехать на экскурсию. Для  $i$ -ой группы определены время начала экскурсии  $s_i$  и завершения экскурсии  $f_i$  (оба времени включительно). Чтобы группе туристов поехать на экскурсию, нужен один автобус. Группа занимает его полностью. Но один автобус может отвезти несколько групп туристов, если их время экскурсий не пересекается. Определите минимальное число автобусов, которое нужно заказать отелю, для того чтобы обслужить все группы туристов.

**Входные данные**

В первой строке задано целое число  $n$  – количество групп туристов. Далее в  $n$  строках пары целых чисел  $s_i, f_i$  через пробел – время начала и конца экскурсии.

$1 \leq n \leq 1000$ .  
 $0 \leq s_i \leq f_i \leq 10^9$   
 (времена заданы в виде целых чисел, в секундах)

**Выходные данные**

Выведите минимальное число автобусов.

**Пример**

INPUT	OUTPUT
5	3
3 6	
4 7	
1 3	
8 9	
1 10	

**Задача № 3 (20 баллов)**

Астрономы часто изучают карты звезд, где звезды представлены точками на плоскости, и каждая звезда имеет декартовы координаты  $(x_i, y_i)$ . На карте находится  $N$  звезд. Астрономы называют уровнем звезды количество звезд, которые не выше и не правее этой звезды. Астрономы хотят знать распределение уровней звезд. Напишите программу, которая считает количество звезд каждого из уровней от 0 до  $N-1$ .

**Входные данные**

В первой строке находится одно целое число  $N$  - количество звезд.

В каждой из следующих  $N$  строках находятся по два целых числа  $x_i, y_i$  - координаты звезд.

$1 \leq N \leq 1000$   
 $0 \leq X \leq 1000$   
 $0 \leq Y \leq 1000, i=1, \dots, N$

Гарантируется, что в одной точке не находятся две звезды.

**Выходные данные**

Выведите  $N$  чисел каждое в отдельную строку, где  $i$ -ое число - это количество звезд  $(i-1)$ -ого уровня.

**Пример**

INPUT	OUTPUT
5	1
1 1	2
5 1	1
7 1	1
3 3	0
5 5	

**Пояснение**

У звезды с координатами  $(1,1)$  уровень 0, потому что нет звезд не выше и не правее ее. У звезды  $(5,1)$  уровень 1, есть только одна звезда  $(1,1)$  не выше и не правее ее. У звезды  $(7,1)$  уровень 2, есть две звезды  $(1,1)$  и  $(5,1)$  не выше и не правее ее. У звезды  $(3,3)$  уровень 1, есть одна звезд  $(1,1)$  не выше и не правее ее. У звезды  $(5,5)$  уровень 3, есть три звезды  $(1,1)$ ,  $(3,3)$  и  $(5,1)$  не

выше и не правее ее. Итого уровень 0 у 1-ой звезды, уровень 1 у 2-х звезд, уровень 2 у 1-ой звезды, уровень 3 у 1-ой звезды, и уровень 4 у 0 звезд.

#### Задача № 4 (20 баллов)

Василий проверяет скобочные последовательности. Правильной скобочной последовательностью называется такая строка из открывающихся и закрывающихся скобок, в которую можно подставить символы 1 и + так, чтобы получилось верное арифметическое выражение. Например,  $((()))$  - правильная скобочная последовательность. Потому что можно составить арифметическое выражение  $1+((1+1)+(1+1))$ . А вот  $(())$  неправильная скобочная последовательность.

У Василия есть строка  $s$  состоящая только из круглых скобок. Помогите Василию найти максимальную длину префикса строки  $s$ , являющегося правильной скобочной последовательностью. Префиксом строки называется часть строки, которая начинается с первого символа и заканчивается на каком-то другом символе. Формально, если строка состоит из символов  $s_1...s_n$ , то префикс это  $s_1...s_i$  для некоторого  $i$ . Сама строка тоже является префиксом самой себя.

#### Входные данные

Единственная строка содержит строку  $s$ . Длина строки не превышает 5000.

#### Выходные данные

Выведите максимальную длину префикса, являющегося правильной скобочной последовательностью. Если таких префиксов нет, то выведите 0.

#### Пример

INPUT	OUTPUT
$((((()))))$	10
$))))((((((($	0
$((())$	6

#### Пояснение к первому примеру

Есть два префикса, являющихся правильной скобочной последовательностью,  $()$  и  $((((()))))$ . При этом у второго из них длина больше.

#### Пояснение к третьему примеру

Вся строка является правильной скобочной последовательностью, она же является префиксом, поэтому ответ — длина строки.

#### Задача № 5 (20 баллов)

Фаниль собирается отправиться к бабушке в деревню. Деревня бабушки Фаниля очень далеко от Казани. Возможно, ему придётся ехать на нескольких автобусах. Фаниль знает информацию о всех  $m$  автобусах и  $n$  станциях в округе. Каждый  $i$ -ый автобус едет без остановок от станции  $a_i$  до станции  $b_i$  и путь этот длины  $r_i$ . Автобус едет только в одном направлении. Длиной пути от Казани до деревни Фаниль называет суммарную длину всех путей автобусов, на которых он ехал. При этом если Фаниль доехал до какой-нибудь остановки, он на ней же должен сесть на следующий автобус, пока не доберется до деревни. Помогите Фанилю найти длину кратчайшего пути из Казани до деревни бабушки. Если таких путей несколько, выберите путь, который состоит из максимального количества пересадок, чтобы Фаниль смог полюбоваться красотой каждой станции.

#### Входные данные

В первой строке заданы два целых числа  $n$  и  $m$  — количество станций и автобусов, которые могут быть полезны для поездки Фаниля. Во второй строке заданы два целых числа  $a$  и  $b$ ,  $a$  —

номер станции в Казани, и  $b$  – номер станции в деревне у бабушки Фаниля ( $1 \leq a, b \leq n$ ). В следующих  $m$  строках заданы тройки целых положительных чисел  $a_i, b_i, p_i$ , описывающие станции, которые соединены автобусным маршрутом, и длина пути между ними. Имейте в виду, что автобус едет только в одну сторону.

$$1 \leq n \leq 100$$

$$1 \leq m \leq 1000$$

$$1 \leq p_i \leq 1000$$

### Выходные данные

В первой строке выведите единственное целое число – длину кратчайшего пути из Казани (станция с номером  $a$ ) до деревни бабушки Фаниля (станция с номером  $b$ ). В следующей строке выведите путь – номера станций по очереди через пробел, через которые едет Фаниль в порядке следования. Гарантируется, что такой путь существует. Если таких путей несколько, выведите путь, который содержит в себе максимальное число станций.

### Пример

INPUT	OUTPUT
5 6	9
1 5	1 3 4 5
1 2 2	
1 3 3	
2 4 3	
3 4 1	
4 5 5	
1 5 9	

### Пояснение

Есть два кратчайших пути длины 9: это путь через 1 и 5 и путь через станции 1, 3, 4 и 5. Необходимо выдать второй из них, так как в нем больше станций.

**Межрегиональные предметные олимпиады КФУ**  
**профиль «Информатика»**  
**заключительный этап**  
**2020-2021 учебный год**  
**Разбор задач**

**Задача №1**

Для решения данной задачи используется такая структура данных, как упорядоченное бинарное дерево. На языке C++ такой структуре может соответствовать тип данных multiset (обычный set не подойдет, так как цены могут быть одинаковыми). Такая структура позволяет за  $O(\log k)$  получать значение, удалять некоторое значение и добавлять новое во множество размера  $k$ .

В multiset мы будем хранить сладости из собираемого набора. Для каждого набора со склада мы сравниваем максимальное значение в multiset, которое мы получаем из множества за  $O(\log k)$ , \*st.begin() (multiset<int> st) с максимальным значением из набора со склада. Если значение из множества больше, удаляем его во множестве и кладем новое значение, равное максимальному значению со склада.

В конце проходим по множеству и считаем сумму его элементов.

Для заданных ограничений 9 и 10 класса подойдет обычный перебор. Для каждого набора со склада ищем максимальное значение в массиве нового набора, заменяем максимальное значение, если оно больше, чем максимальное значение  $i$ -ого набора. В конце считаем сумму элементов массива.

Для ограничений 9 и 10 класса подойдет моделирование ситуации на «быстрых языках программирования» (C++, например). Сложность  $O(N*k)$ .

Для 10 и 11 класса сумма элементов должна была храниться в 8-байтном целочисленном типе данных (например, int64, long long и т.д.).

Разбалловка

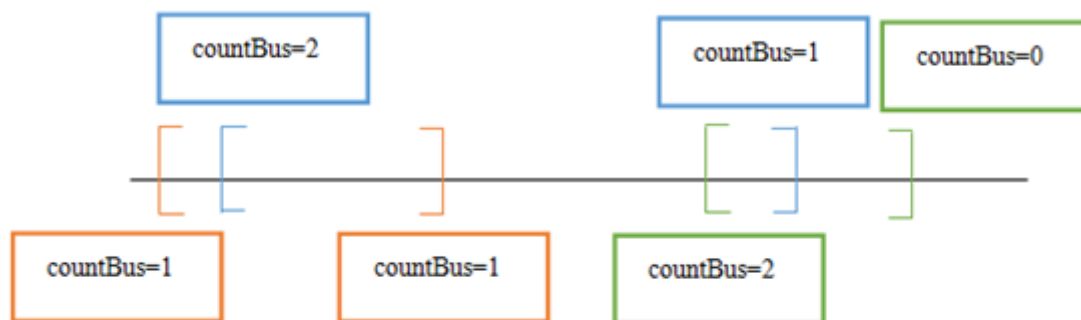
Пояснение	Балл
Есть неверный код с направлением в правильное решение, правильно сработавший на одном-двух тестах	1.
Есть неверный код с направлением в правильное решение, правильно сработавший на трёх-пяти тестах	2.
Кратко описана идея алгоритма без самого алгоритма	3.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по времени. Есть ошибки реализации	4.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по времени.	5.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по времени. Однако есть идеи, которые могут ускорить алгоритм.	6.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям. Но программа не удовлетворяет ограничениям по времени, на котором программа будет работать неверно. Однако есть идеи, которые могут ускорить алгоритм и сократить используемую память.	7.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по памяти. Есть не более пяти ошибок в реализации.	8.

Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по памяти. Есть не более двух ошибок в реализации.	9.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по памяти.	10.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с грубыми ошибками в реализации.	11.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, однако используются функции без подключения библиотек (например, для функции сортировки в c++ необходимо подключить <code>algorithm</code> )	12.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более пяти)	13.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более трёх)	14.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более двух)	15.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более одной)	16.
Программа корректная, удовлетворяет ограничениям по времени и по памяти, однако неправильно задано одно условие или значения цикла (пример: должно начинаться с 0, а в программе начинается с 1), или есть выход за границы массива. (2 условия вместе). Программа неправильно работает на не более трех тестов.	17.
Программа корректная, удовлетворяет ограничениям по времени и по памяти, однако неправильно задано <b>одно</b> условие или значения цикла (пример: должно начинаться с 0, а в программе начинается с 1), или есть выход за границы массива. Программа неправильно работает на не более двух тестов.	18.
Написана корректная программа, удовлетворяющая ограничениям по времени и памяти. Нельзя придумать тест, соответствующий основным ограничениям входных данных, на которых будет неверный ответ. Однако есть несущественная ошибка: не рассмотрен частный случай (неправильно рассмотрен). Программа неправильно работает на тесте с частным случаем.	19.
Написана корректная программа, удовлетворяющая ограничениям по времени и памяти. Нельзя придумать тест, соответствующий ограничениям входных данных, на которых будет неверный ответ	20.

### Задача №2

Сохраняем значения всех времен в одном массиве с отметкой, какое время является началом экскурсии, а какой концом. Например, можно использовать структуру данных `pair` в C++, где первое значение – значение времени, второе – 0 (начало экскурсии) или 1 (конец экскурсии). На языке Pascal можно описать свою структуру данных через `type – record`.

Сортируем массив пар по возрастанию. Проходимся по массиву: если какая-то экскурсия началась, увеличиваем количество автобусов, если закончилась, уменьшаем количество автобусов. Ответом будет максимальное значение автобусов за всё время.



По изображению мы видим, что максимальное количество автобусов, которое было за всё время равно 2, это и будет ответом. Автобус, который отвез оранжевую группу на экскурсии, сможет отвезти и зеленую группу на экскурсию.

Для ограничений 9 и 10 класса можно использовать пузырьковую сортировку. Ограничения 11 класса требуют использовать сортировки, работающие за  $O(N \log N)$ .

#### Разбалловка

Пояснение	Балл
Есть неверный код с направлением в правильное решение, правильно сработавший на одном-двух тестах	1.
Есть неверный код с направлением в правильное решение, правильно сработавший на трёх-пяти тестах	2.
Кратко описана идея алгоритма без самого алгоритма	3.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по времени. Есть ошибки реализации	4.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по времени.	5.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по времени. Однако есть идеи, которые могут ускорить алгоритм.	6.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям. Но программа не удовлетворяет ограничениям по времени, на котором программа будет работать неверно. Однако есть идеи, которые могут ускорить алгоритм и сократить используемую память.	7.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по памяти. Есть не более пяти ошибок в реализации.	8.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по памяти. Есть не более двух ошибок в реализации.	9.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно.	10.

Но программа не удовлетворяет ограничениям по памяти.	
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с грубыми ошибками в реализации.	11.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, однако используются функции без подключения библиотек (например, для функции сортировки в c++ необходимо подключить algorithm)	12.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более пяти)	13.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более трёх)	14.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более двух)	15.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более одной)	16.
Программа корректная, удовлетворяет ограничениям по времени и по памяти, однако неправильно задано одно условие или значения цикла (пример: должно начинаться с 0, а в программе начинается с 1), или есть выход за границы массива. (2 условия вместе). Программа неправильно работает на не более трех тестов.	17.
Программа корректная, удовлетворяет ограничениям по времени и по памяти, однако неправильно задано <b>одно</b> условие или значения цикла (пример: должно начинаться с 0, а в программе начинается с 1), или есть выход за границы массива. Программа неправильно работает на не более двух тестов.	18.
Написана корректная программа, удовлетворяющая ограничениям по времени и памяти. Нельзя придумать тест, соответствующий основным ограничениям входных данных, на которых будет неверный ответ. Однако есть несущественная ошибка: не рассмотрен частный случай (неправильно рассмотрен). Программа неправильно работает на тесте с частным случаем.	19.
Написана корректная программа, удовлетворяющая ограничениям по времени и памяти. Нельзя придумать тест, соответствующий ограничениям входных данных, на которых будет неверный ответ	20.

### Задача №3

Объявим массив целых чисел  $Levels[N]$ , где нумерация будет от 0 до  $N-1$ , и обнулیم его. Храним все точки в массиве пар. Сортируем массив по значению  $X$ . Для каждой  $i$ -ой звезды, в отрезке от 0 до  $i-1$  проверяем значения  $y_i \geq y_j$  для  $j = [1, \dots, i-1]$ , считаем количество таких  $y_j$ , записываем полученное количество в  $k$ .  $Levels[k]$  увеличиваем на 1. И так для каждой звезды. Сложность  $O(N^2)$ .

Для ограничений всех классов подойдет переборный алгоритм. Нужно для каждой  $i$ -ой звезды проверять условия, что звезда  $j$  не правее и не выше, а также  $j$  не равно  $i$ .

Членами жюри, проверяющими данную задачу было выделено 3 типа ошибок/



Тип ошибки	Балл
Отсутствуют ошибки. Представлен рабочий верный код.	20
Сравнение точки $i$ с точкой $i$ .	19
Сравниваются не все пары, при этом предварительной сортировки не было.	11
Работает на ограниченном количестве тестов.	5

#### Задача №4

Поддерживаем разность открывающихся и закрывающихся скобок. Если у нас разность стала отрицательна, то ответ текущая позиция  $-1$ . Если дошли до конца и разность не была отрицательна и сейчас она не  $0$ , то тогда нет ответа (ответ  $0$ ). Если дошли до конца строки и разность равна  $0$ , тогда ответ – длина строки.

Большинство участников искали максимальную правильную подстроку в строке. Это неверно. Задача состояла в поиске правильного префикса.

#### Разбалловка

Пояснение	Балл
Есть неверный код с направлением в правильное решение, правильно сработавший на одном-двух тестах	1.
Есть неверный код с направлением в правильное решение, правильно сработавший на трёх-пяти тестах	2.
Кратко описана идея алгоритма без самого алгоритма	3.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по времени. Есть ошибки реализации	4.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по времени.	5.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по времени. Однако есть идеи, которые могут ускорить алгоритм.	6.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям. Но программа не удовлетворяет ограничениям по времени, на котором программа будет работать неверно. Однако есть идеи, которые могут ускорить алгоритм и сократить используемую память.	7.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по памяти. Есть не более пяти ошибок в реализации.	8.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по памяти. Есть не более двух ошибок в реализации.	9.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по памяти.	10.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с грубыми ошибками в реализации.	11.

Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, однако используются функции без подключения библиотек (например, для функции сортировки в c++ необходимо подключить algorithm)	12.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более пяти)	13.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более трёх)	14.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более двух)	15.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более одной)	16.
Программа корректная, удовлетворяет ограничениям по времени и по памяти, однако неправильно задано одно условие или значения цикла (пример: должно начинаться с 0, а в программе начинается с 1), или есть выход за границы массива. (2 условия вместе). Программа неправильно работает на не более трех тестов.	17.
Программа корректная, удовлетворяет ограничениям по времени и по памяти, однако неправильно задано <b>одно</b> условие или значения цикла (пример: должно начинаться с 0, а в программе начинается с 1), или есть выход за границы массива. Программа неправильно работает на не более двух тестов.	18.
Написана корректная программа, удовлетворяющая ограничениям по времени и памяти. Нельзя придумать тест, соответствующий основным ограничениям входных данных, на которых будет неверный ответ. Однако есть несущественная ошибка: не рассмотрен частный случай (неправильно рассмотрен). Программа неправильно работает на тесте с частным случаем.	19.
Написана корректная программа, удовлетворяющая ограничениям по времени и памяти. Нельзя придумать тест, соответствующий ограничениям входных данных, на которых будет неверный ответ	20.

### Задача №5

Для решения задачи используем алгоритм Дейкстры. Для каждой вершины храним, количество вершин, из которых мы пришли в эту вершину. Пусть массив  $P[N]$  будет хранить количество таких вершин-родителей. Пусть массив  $Par[N]$  – массив родителей.  $Par[i]$  – номер вершины, из которой мы пришли в вершину  $i$ . Для вершины  $a$   $P[a]=0$ . Когда мы производим релаксацию по ребрам, увеличиваем  $P[e.to]=P[v]+1$ , где  $v$  – выбранная вершина с минимальным расстоянием, а  $e$  ребро, исходящее из вершины  $v$ ,  $e.to$  – номер смежной  $v$  вершины.

При выборе минимальной вершины  $v$ , выбираем ту, у которой количество вершин-родителей больше:

for  $j \in V$

if !used[j] and ( $v == \text{null}$  or  $d[j] < d[v]$  or  $d[j]==d[v]$  and  $P[j]>P[v]$ )

$v = j$

for  $e$  : исходящие из  $v$  рёбра

if  $d[v] + e.len < d[e.to]$

$d[e.to] = d[v] + e.len$

$$P[e.to] = P[v] + 1$$

$$Par[e.to] = v$$

Длина пути будет сохранена в  $d[b]$ . Путь можно восстановить рекурсивно или итеративно с помощью массива  $Par$ .

Некоторые участники не учитывали, что цикл в графе не исключен. Классические алгоритмы  $dfs$  и  $bfs$  либо не проверяли все пути, либо закикливались. Таки алгоритмы считаются неверными решениями.

#### Разбалловка

Пояснение	Балл
Есть неверный код с направлением в правильное решение, правильно сработающий на одном-двух тестах	1.
Есть неверный код с направлением в правильное решение, правильно сработающий на трёх-пяти тестах	2.
Кратко описана идея алгоритма без самого алгоритма	3.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по времени. Есть ошибки реализации	4.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по времени.	5.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по времени. Однако есть идеи, которые могут ускорить алгоритм.	6.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям. Но программа не удовлетворяет ограничениям по времени, на котором программа будет работать неверно. Однако есть идеи, которые могут ускорить алгоритм и сократить используемую память.	7.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по памяти. Есть не более пяти ошибок в реализации.	8.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по памяти. Есть не более двух ошибок в реализации.	9.
Решение корректное, нельзя придумать тест, удовлетворяющий ограничениям, на котором программа будет работать неверно. Но программа не удовлетворяет ограничениям по памяти.	10.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с грубыми ошибками в реализации.	11.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, однако используются функции без подключения библиотек (например, для функции сортировки в $c++$ необходимо подключить <code>algorithm</code> )	12.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более пяти)	13.
Правильная идея алгоритма, удовлетворяющего ограничениям	14.

по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более трёх)	
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более двух)	15.
Правильная идея алгоритма, удовлетворяющего ограничениям по времени и памяти, с небольшими ошибками в реализации (код не скомпилируется) (не более одной)	16.
Программа корректная, удовлетворяет ограничениям по времени и по памяти, однако неправильно задано одно условие или значения цикла (пример: должно начинаться с 0, а в программе начинается с 1), или есть выход за границы массива. (2 условия вместе). Программа неправильно работает на не более трех тестов.	17.
Программа корректная, удовлетворяет ограничениям по времени и по памяти, однако неправильно задано <b>одно</b> условие или значения цикла (пример: должно начинаться с 0, а в программе начинается с 1), или есть выход за границы массива. Программа неправильно работает на не более двух тестов.	18.
Написана корректная программа, удовлетворяющая ограничениям по времени и памяти. Нельзя придумать тест, соответствующий основным ограничениям входных данных, на которых будет неверный ответ. Однако есть несущественная ошибка: не рассмотрен частный случай (неправильно рассмотрен). Программа неправильно работает на тесте с частным случаем.	19.
Написана корректная программа, удовлетворяющая ограничениям по времени и памяти. Нельзя придумать тест, соответствующий ограничениям входных данных, на которых будет неверный ответ	20.