Радо-стные графы (7 класс) / *Rado*-inspired graphs (7th degree)

Рассмотрим непустое множество V точек и множество E линий, соединяющих некоторые (возможно, никакие, а возможно, что все) пары этих точек. Тогда будем называть пару (V, E) **графом**, элементы V – **вершинами** графа, а элементы E – его **рёбрами**.

Нас не будут интересовать геометрические свойства графа: длины рёбер, углы между ними, наличие или отсутствие точек их пересечения и т.д. – важны только набор вершин и то, какие из них соединены рёбрами. Направление ребра нам тоже не будет важно, то есть далее будем рассматривать только так называемые неориентированные графы. При этом давайте условимся, что каждое ребро соединяет две различные вершины (т.е. в графе нет петель), и любые две вершины соединены не более чем одним ребром (т.е. в графе нет кратных рёбер) – такие графы называют простыми. Далее будем рассматривать только простые графы.

Если между двумя любыми вершинами графа можно пройти по его рёбрам единственным способом (не проходя никакую вершину дважды), то такой граф называется **деревом**.

Допустим, определен граф (V, E). Тогда его **подграфом** будем называть такой граф, который можно получить из графа (V, E) удалением некоторых (возможно, ни одной) вершин и всех выходящих из них рёбер.

Представим, что у нас есть два графа – (V_1, E_1) и (V_2, E_2). Будем называть эти графы **изоморфными**, если можно установить такое соответствие f между их вершинами, что

- для каждой вершины $v_1 \in V_1$ $f(v_1) = v_2 \in V_2$, и наоборот, т.е. для каждой вершины $v_2 \in V_2$ найдется такая вершина $v_1 \in V_1$, что $f(v_1) = v_2$;
- если вершины $v_1, v_2 \in V_1$ соединены ребром, то и вершины $v_3 = f(v_1)$, $v_4 = f(v_2)$ $(v_3, v_4 \in V_2)$ тоже соединены ребром; и наоборот, т.е. если вершины $v_3, v_4 \in V_2$ соединены ребром, то найдутся такие $v_1, v_2 \in V_1$, которые тоже соединены ребром, и для которых $v_3 = f(v_1)$, $v_4 = f(v_2)$.

Будем говорить, что граф (V_2, E_2) **вложен** в граф (V_1, E_1), если граф (V_2, E_2) изоморфен некоторому подграфу графа (V_1, E_1).

Consider a non-empty set of points V and a set of lines E connecting some (possibly none, and possibly all) of these pairs of points. Then we will call the pair (V, E) a **graph**, the elements of V are the **vertices** of the graph, and the elements of E are its **edges**.

We will not be interested in geometric properties of the graph: the length of the edges, the angles between them, the presence or absence of their intersection points, etc. – all that matters is the set of vertices and which of them are connected by the edges. The direction of the edge will also not be important to us, so from now on we will consider only so-called undirected graphs. Let's also agree that each edge connects two different vertices (i.e. there are no loops in the graph), and any two vertices are connected by at most one edge (i.e. there are no multi-edges in the graph) – such graphs are called simple. From now on, we will consider only simple graphs.

If between any two vertices of a graph one can walk along its edges in a unique way (without passing any vertex twice), then the graph is called a **tree**.

Lets consider some graph (V, E). Then we call its **subgraph** a graph that can be obtained from the graph (V, E) by removing some (possibly none) vertices and all edges coming from them.

Lets consider that we have two graphs – (V_1, E_1) and (V_2, E_2). We will call these graphs **isomorphic** if we can establish a correspondence f between their vertices such that

- for each vertex $v_1 \in V_1$ $f(v_1) = v_2 \in V_2$, and vice versa, i.e. for each vertex $v_2 \in V_2$ a vertex $v_1 \in V_1$ exists such that $f(v_1) = v_2$;
- if the vertices $v_1, v_2 \in V_1$ are connected by an edge, then the vertices $v_3 = f(v_1)$, $v_4 = f(v_2)$ $(v_3, v_4 \in V_2)$ are also connected by an edge; and vice versa, i.e. if the vertices $v_3, v_4 \in V_2$ are connected by an edge, then there are $v_1, v_2 \in V_1$ that are also connected by an edge, and they satisfy $v_3 = f(v_1)$, $v_4 = f(v_2)$.

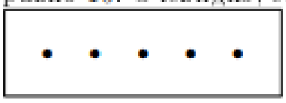
We will say that the graph (V_2, E_2) is **embedded** in the graph (V_1, E_1) if the graph (V_2, E_2) is isomorphic to some subgraph of the graph (V_1, E_1).

Задача 1. (15 points)
Сколько существует различных (т.е. неизоморфных друг другу) графов, содержащих по 5 вершин?

How many different (i.e., non-isomorphic to each other) graphs with 5 vertices are there?

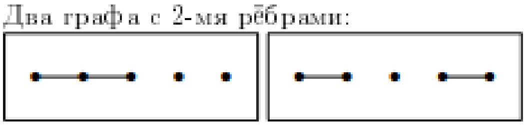
Решение. Вообще, вычисление количества неизоморфных простых графов с заданным количеством вершин – задача на удивление непростая. Поэтому мы решим ее упорядоченным перебором, тем более что в наших графах будет всего по 5 вершин.

Можно составить $\frac{1}{2} \cdot 5 \cdot 4 = 10$ пар из 5 вершин, т.е. максимальное возможное количество рёбер равно 10. Очевидно, есть только один граф с 0 рёбер и 5 вершинами:

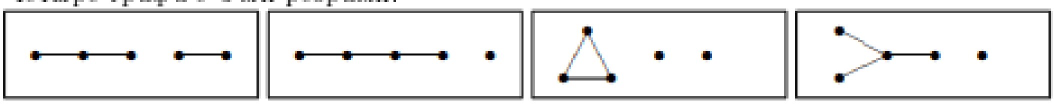


Также существует лишь один граф (здесь и далее – с точностью до изоморфизма) с 1 ребром:

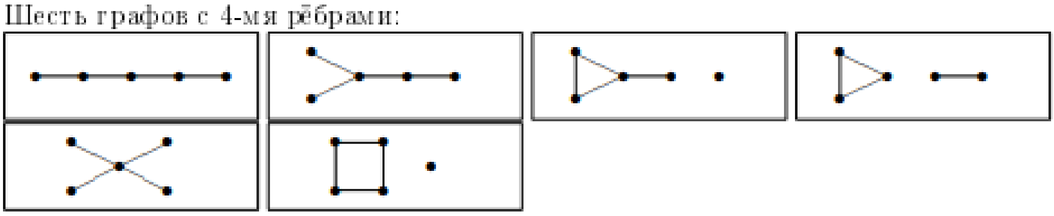
Два графа с 2-мя рёбрами:



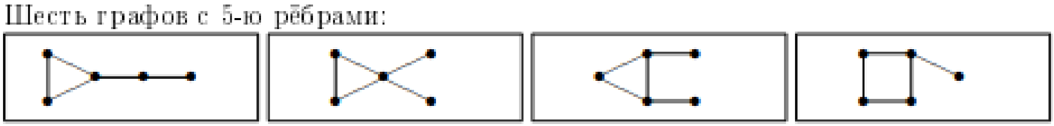
Четыре графа с 3-мя рёбрами:

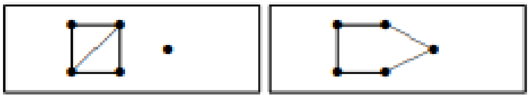


Шесть графов с 4-мя рёбрами:



Шесть графов с 5-ю рёбрами:





Графы с 6-ю (и более) вершинами рассматривать не потребуется, поскольку они получаются из т.н. полного графа (т.е. содержащего все возможные рёбра, в нашем случае – 10 рёбер) удалением рёбер, соответствующих уже перечисленным графам. Так, например, графов с 2-мя рёбрами столько же, сколько графов с $10 - 2 = 8$-ю рёбрами.

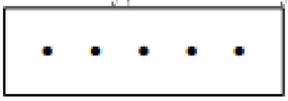
Итак, искомое число равно $2 \cdot (1 + 1 + 2 + 4 + 6) + 6 = 34$.

Критерии оценивания:

- ответ неверный – 0 баллов;
- ответ верный – 15 баллов.

Solution (ENG). By the way, calculating the number of non-isomorphic simple graphs with a given number of vertices is a surprisingly difficult task. Therefore, we will solve it using an ordered search, especially since our graphs will only have 5 vertices.

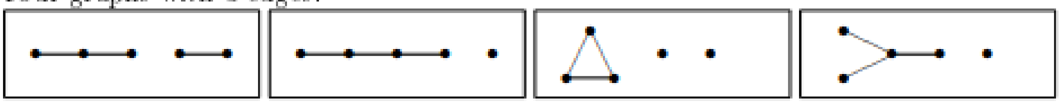
You can make $\frac{1}{2} \cdot 5 \cdot 4 = 10$ pairs of 5 vertices, i.e. the maximum possible number of edges is 10. Obviously, there is only one graph with 0 edges and 5 vertices:



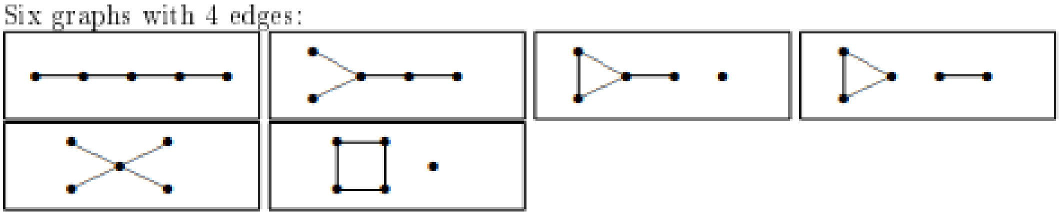
There is also only one graph (hereinafter – up to isomorphism) with 1 edge:

Two graphs with 2 edges:

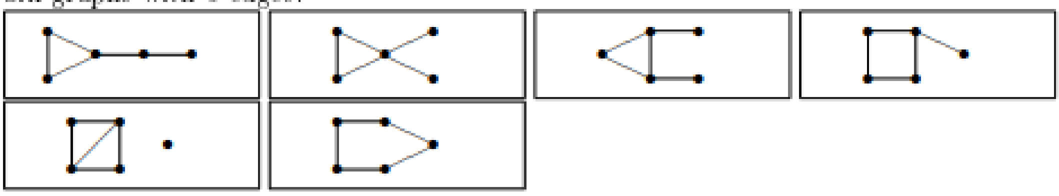
Four graphs with 3 edges:



Six graphs with 4 edges:



Six graphs with 5 edges:



Graphs with 6 (or more) vertices do not need to be considered, since they are obtained from the so-called. a complete graph (i.e. containing all possible edges, in our case 10 edges) by removing the edges corresponding to the ones from already listed graphs. So, for example, there are as many graphs with 2 edges as there are with $10 - 2 = 8$ edges.

By that, the required number is $2 \cdot (1 + 1 + 2 + 4 + 6) + 6 = 34$.

Criteria:

- the given answer is incorrect – 0 points;
- the given answer is correct - 15 points.

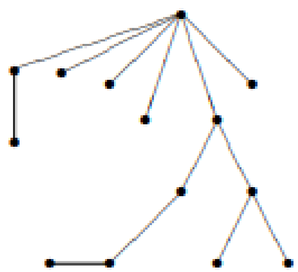
Answer: 34

Задача 2. (20 points)
Приведите пример дерева G, которое содержит все возможные неизоморфные друг другу деревья из 7 вершин в качестве вложенных.

Give an example of a tree G that contains all possible non-isomorphic trees of 7 vertices as its embedded ones.

Решение.

Solution (ENG).



Критерии оценивания:

- нет решения, либо оно неразборчиво – 0 баллов;
- в приведенное дерево вложены некоторые деревья на 7 вершинах – 5 баллов;
- в приведенное дерево вложены многие деревья на 7 вершинах – 10 баллов;
- в приведенное дерево вложены почти все (за исключением одного-двух) деревья на 7 вершинах – 15 баллов;
- в приведенное дерево вложены все деревья на 7 вершинах – 20 баллов;

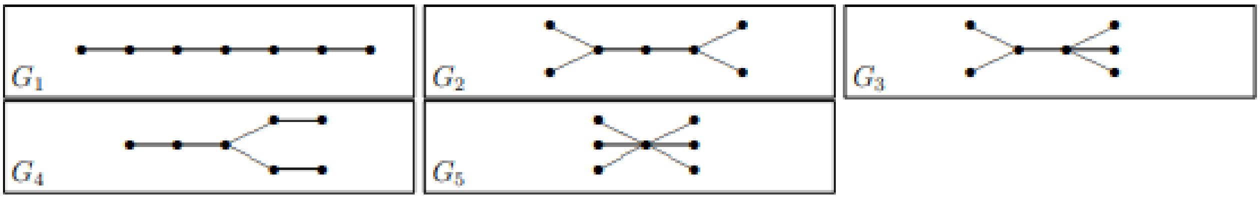
Criteria:

- there is no solution or it is illegible – 0 points;
- the given tree contains some trees with 7 vertices – 5 points;
- the given tree contains many trees with 7 vertices – 10 points;
- almost all (with the exception of one or two) trees on 7 vertices are embedded in the given tree – 15 points;
- the given tree contains all the trees on 7 vertices – 20 points;

Задача 3. (25 points)
Какое наименьшее число вершин может содержать дерево G, описанное в предыдущей задаче? Ответ обоснуйте.

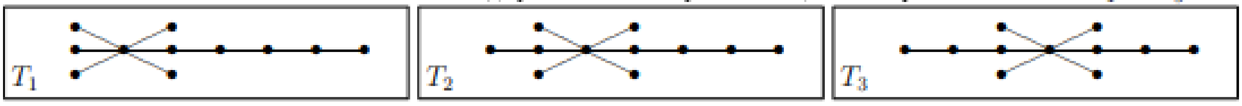
What is the smallest number of vertices that the tree G from the previous task can contain? Explain your answer.

Решение. Существуют 11 попарно неизоморфных деревьев на 7-и вершинах, среди которых есть следующие:



Пример дерева G на 14 вершинах приведен в решении задачи 2; осталось доказать, что никакое дерево на 13 вершинах не удовлетворяет условию (очевидно, если ему удовлетворяет какое-то дерево с $k < 13$ вершинами, то добавлением к нему $(13 - k)$ вершин без рёбер также получим требуемое дерево).

Во-первых, G должно содержать «цепь» G_1 из 7 вершин, и в тоже время содержать дерево G_5, в котором одна из вершин соединена с 6-ю остальными – значит, общее число вершин G не может быть меньше 11. Вот все возможные деревья с 11 вершинами, в которые вложены G_1 и G_5:

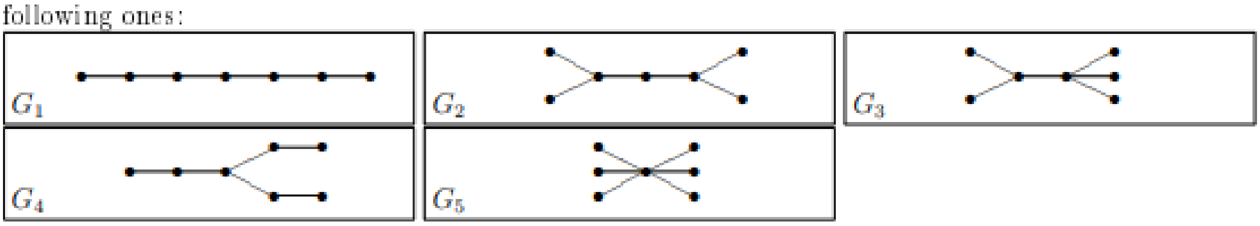


В деревьях G_2, G_3 есть по 2 вершины степени не меньше 3, причем в G_2 эти вершины соединены ребром, а в G_3 – нет, поэтому к любому из деревьев T_1, T_2, T_3 необходимо добавить минимум по 2 вершины к разным вершинам степени 2 – итого в G не менее 13 вершин. В то же время в дереве G_4 вершина степени 3 соединена с двумя цепями, причем каждая из этих цепей состоит из двух вершин – легко проверить, что ни одно из упомянутых деревьев на 13 вершинах не содержит G_4 в качестве вложенного, а значит, необходимо добавить еще как минимум 1 вершину, т.е. в G не менее 14 вершин, что и требовалось доказать.

Критерии оценивания:

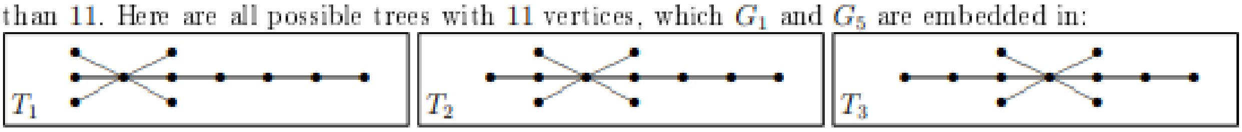
- нет решения (в т.ч. если приведен только ответ) – 0 баллов;
- даны ответ и пример – 10 баллов;
- приведена верная оценка – 15 баллов;
- приведены верные оценка и пример – 25 баллов.

Solution (ENG). There are 11 pairwise non-isomorphic trees on 7 vertices, among which are the following ones:



An example of G – the required tree on 14 vertices – is given in the solution of task 2; it remains to prove that no tree with 13 vertices satisfies the condition (obviously, if some tree with $k < 13$ vertices satisfies it, then by adding $(13 - k)$ vertices without edges to it we will also obtain the required tree).

First, G must contain a «chain» G_1 of 7 vertices, also it must contain a tree G_5, in which one of the vertices is connected to 6 others, which means that the total number of vertices is G cannot be less than 11. Here are all possible trees with 11 vertices, which G_1 and G_5 are embedded in:



The trees G_2, G_3 each have 2 vertices of degree at least 3, and in G_2 these vertices are connected by an edge, but in G_3 they are not, so to any of the trees T_1, T_2, T_3 it is necessary to add a minimum of 2 vertices to different vertices of degree 2 – by that, there are at least 13 vertices in G. At the same time, in the tree G_4, a vertex of degree 3 is connected to two chains with each of them consisting of two vertices – it is easy to check that none of the mentioned trees with 13 vertices contains G_4 as embedded one, so it is necessary to add at least 1 more vertex, i.e. there are at least 14 vertices in G, which is what we needed to prove.

Criteria:

- there is no solution (or if only the answer is given) – 0 points;
- the answer and example are given – 10 points;
- the correct estimation is given – 15 points;
- the correct estimation and example are given – 25 points.

Answer: 14

Задача 4. (40 points)
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 Мб

У Никиты было два одинаковых неориентированных графа из n вершин и m ребер, вершины графов пронумерованы различными целыми числами от 1 до n. Так же известно, что степень каждой вершины не превосходит 2.

Ильнуру не понравилось, что графы одинаковые, и он перенумеровал номера вершин первого графа произвольным образом. В результате перенумерации, номера вершин первого графа все еще являются различными целыми числами от 1 до n.

Вам даны итоговые два графа. Требуется проверить, могли ли эти графы получиться в результате перенумерации, и перенумеровать вершины первого графа обратно, чтобы получились одинаковые графы.

Input file:
Первая строка содержит два целых числа n и m ($1 \leq n \leq 8$, $0 \leq m \leq \frac{n(n-1)}{2}$) — количество вершин и ребер графов соответственно.

Каждая из следующих m строк содержит два целых числа u_i и v_i ($1 \leq u_i < v_i \leq n$) — ребра первого графа.

Следующие m строк задают второй граф в формате, аналогичном первому графу.

Гарантируется, что у каждого из двух графов все ребра попарно различны и степень каждой вершины не превосходит 2.

Output file:
Если графы не могли получиться в результате перенумерации, выведите «NO».

Иначе, в первой строке выведите «YES».

На следующей строке выведите n различных целых чисел p_i от 1 до n, означающих, что вершина с номером i первого графа должна получить номер p_i. В результате такой перенумерации должны получиться одинаковые графы.

Вы можете выводить каждую букву в любом регистре (строчную или заглавную). Например, строки «yEs», «yes», «Yes» и «YES» будут приняты как положительный ответ.

Критерии оценивания: +2 балла за каждый пройденный тест. Общее количество тестов – 20.

Example:
Input:
3 2
1 2
2 3
1 3

2 3
Output:
YES
1 3 2

В примере нужно переименовать вершину с номером 2 в 3, и вершину с номером 3 в 2, чтобы получить одинаковые графы.

Time limit: 1 second
Memory usage limit: 256 MB

Nikita had two identical undirected graphs of n vertices and m edges; the vertices of the graphs were numbered by different integers from 1 to n. It is also known that the degree of each vertex does not exceed 2.

Ilnur didn't like that the graphs were the same, and he renumbered the numbers of the vertices of the first graph in an arbitrary way. As a result of renumbering, the vertex numbers of the first graph are still distinct integers from 1 to n.

You are given the final two columns. It is required to check whether these graphs could have been obtained as a result of renumbering, and to renumber the vertices of the first graph back so that identical graphs are obtained.

Input file:

The first line contains two integers n and m ($1 \leq n \leq 8$, $0 \leq m \leq \frac{n(n-1)}{2}$) — number of vertices and graph edges, respectively.

Each of the next m lines contains two integers u_i and v_i ($1 \leq u_i < v_i \leq n$) — edges of the first graph.

The next m lines define the second graph in a format similar to the first graph.

It is guaranteed that each of the two graphs has all edges that are pairwise distinct and that the degree of each vertex does not exceed 2.

Output file:

If the graphs could not be obtained as a result of renumbering, output «NO».

Otherwise, print «YES» on the first line.

On the next line print n different integers p_i from 1 to n, meaning that vertex number i of the first graph should be numbered p_i. As a result of such renumbering, identical graphs should be obtained.

You can display each letter in any case (lowercase or uppercase). For example, the strings «yEs», «yes», «Yes» and «YES» will be accepted as a positive response.

Criteria: $+2$ points for passing of each test. The total number of tests is 20.

Example:

Input:
3 2
1 2
2 3
1 3
2 3
Output:
YES
1 3 2

In the example you need to rename vertex number 2 to 3, and vertex number 3 to 2 to get identical graphs.

Радо-стные графы (8-9 классы) / *Rado*-inspired graphs (8-9 degree)

Рассмотрим непустое множество V точек и множество E линий, соединяющих некоторые (возможно, никакие, а возможно, что все) пары этих точек. Тогда будем называть пару (V, E) **графом**, элементы V – **вершинами** графа, а элементы E – его **рёбрами**.

Нас не будут интересовать геометрические свойства графа: длины рёбер, углы между ними, наличие или отсутствие точек их пересечения и т.д. – важны только набор вершин и то, какие из них соединены рёбрами. Направление ребра нам тоже не будет важно, то есть далее будем рассматривать только так называемые неориентированные графы. При этом давайте условимся, что каждое ребро соединяет две различные вершины (т.е. в графе нет петель), и любые две вершины соединены не более чем одним ребром (т.е. в графе нет кратных рёбер) – такие графы называют простыми. Далее будем рассматривать только простые графы.

Допустим, определен граф (V, E). Тогда его **подграфом** будем называть такой граф, который можно получить из графа (V, E) удалением некоторых (возможно, ни одной) вершин и всех выходящих из них рёбер.

Представим, что у нас есть два графа – (V_1, E_1) и (V_2, E_2). Будем называть эти графы **изоморфными**, если можно установить такое соответствие f между их вершинами, что

- для каждой вершины $v_1 \in V_1$ $f(v_1) = v_2 \in V_2$, и наоборот, т.е. для каждой вершины $v_2 \in V_2$ найдется такая вершина $v_1 \in V_1$, что $f(v_1) = v_2$;
- если вершины $v_1, v_2 \in V_1$ соединены ребром, то и вершины $v_3 = f(v_1)$, $v_4 = f(v_2)$ $(v_3, v_4 \in V_2)$ тоже соединены ребром; и наоборот, т.е. если вершины $v_3, v_4 \in V_2$ соединены ребром, то найдутся такие $v_1, v_2 \in V_1$, которые тоже соединены ребром, и для которых $v_3 = f(v_1)$, $v_4 = f(v_2)$.

Будем говорить, что граф (V_2, E_2) **вложен** в граф (V_1, E_1), если граф (V_2, E_2) изоморфен некоторому подграфу графа (V_1, E_1).

Consider a non-empty set of points V and a set of lines E connecting some (possibly none, and possibly all) of these pairs of points. Then we will call the pair (V, E) a **graph**, the elements of V are the **vertices** of the graph, and the elements of E are its **edges**.

We will not be interested in geometric properties of the graph: the length of the edges, the angles between them, the presence or absence of their intersection points, etc. – all that matters is the set of vertices and which of them are connected by the edges. The direction of the edge will also not be important to us, so from now on we will consider only so-called undirected graphs. Let's also agree that each edge connects two different vertices (i.e. there are no loops in the graph), and any two vertices are connected by at most one edge (i.e. there are no multi-edges in the graph) – such graphs are called simple. From now on, we will consider only simple graphs.

Lets consider some graph (V, E). Then we call its **subgraph** a graph that can be obtained from the graph (V, E) by removing some (possibly none) vertices and all edges coming from them.

Lets consider that we have two graphs – (V_1, E_1) and (V_2, E_2). We will call these graphs **isomorphic** if we can establish a correspondence f between their vertices such that

- for each vertex $v_1 \in V_1$ $f(v_1) = v_2 \in V_2$, and vice versa, i.e. for each vertex $v_2 \in V_2$ a vertex $v_1 \in V_1$ exists such that $f(v_1) = v_2$;

- if the vertices $v_1, v_2 \in V_1$ are connected by an edge, then the vertices $v_3 = f(v_1)$, $v_4 = f(v_2)$ ($v_3, v_4 \in V_2$) are also connected by an edge; and vice versa, i.e. if the vertices $v_3, v_4 \in V_2$ are connected by an edge, then there are $v_1, v_2 \in V_1$ that are also connected by an edge, and they satisfy $v_3 = f(v_1)$, $v_4 = f(v_2)$.

We will say that the graph (V_2, E_2) is **embedded** in the graph (V_1, E_1) if the graph (V_2, E_2) is isomorphic to some subgraph of the graph (V_1, E_1).

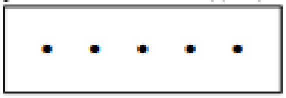
Задача 1. (10 points)

Сколько существует различных (т.е. неизоморфных друг другу) графов, содержащих по 5 вершин?

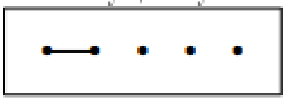
How many different (i.e., non-isomorphic to each other) graphs with 5 vertices are there?

Решение. Вообще, вычисление количества неизоморфных простых графов с заданным количеством вершин – задача на удивление непростая. Поэтому мы решим ее упорядоченным перебором, тем более что в наших графах будет всего по 5 вершин.

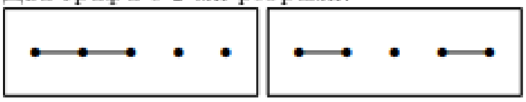
Можно составить $\frac{1}{2} \cdot 5 \cdot 4 = 10$ пар из 5 вершин, т.е. максимальное возможное количество рёбер равно 10. Очевидно, есть только один граф с 0 рёбер и 5 вершинами:



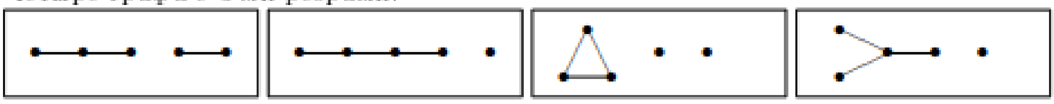
Также существует лишь один граф (здесь и далее – с точностью до изоморфизма) с 1 ребром:



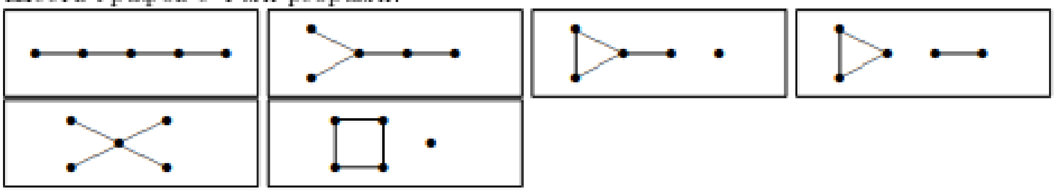
Два графа с 2-мя рёбрами:



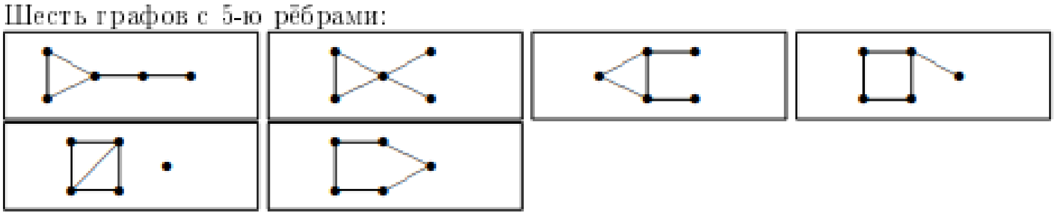
Четыре графа с 3-мя рёбрами:



Шесть графов с 4-мя рёбрами:



Шесть графов с 5-ю рёбрами:



Графы с 6-ю (и более) вершинами рассматривать не потребуется, поскольку они получаются из т.н. полного графа (т.е. содержащего все возможные рёбра, в нашем случае – 10 рёбер) удалением рёбер, соответствующих уже перечисленным графам. Так, например, графов с 2-мя рёбрами столько же, сколько графов с $10 - 2 = 8$-ю рёбрами.

Итак, искомое число равно $2 \cdot (1 + 1 + 2 + 4 + 6) + 6 = 34$.

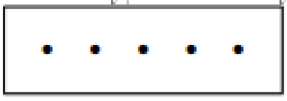
Критерии оценивания:

- ответ неверный – 0 баллов;

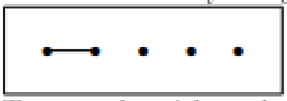
- ответ верный – 10 баллов.

Solution (ENG). By the way, calculating the number of non-isomorphic simple graphs with a given number of vertices is a surprisingly difficult task. Therefore, we will solve it using an ordered search, especially since our graphs will only have 5 vertices.

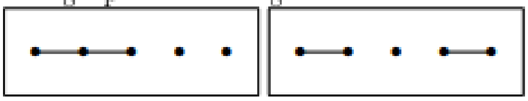
You can make $\frac{1}{2} \cdot 5 \cdot 4 = 10$ pairs of 5 vertices, i.e. the maximum possible number of edges is 10. Obviously, there is only one graph with 0 edges and 5 vertices:



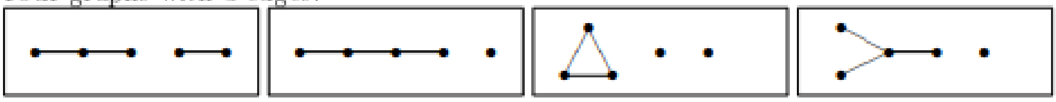
There is also only one graph (hereinafter – up to isomorphism) with 1 edge:



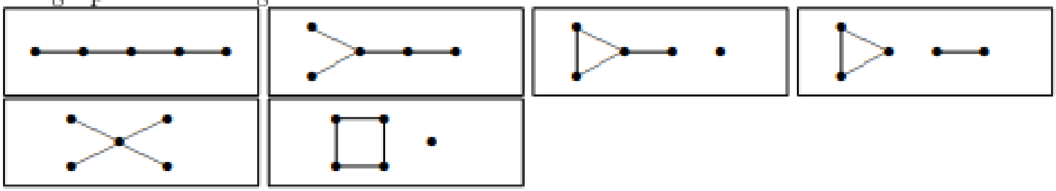
Two graphs with 2 edges:



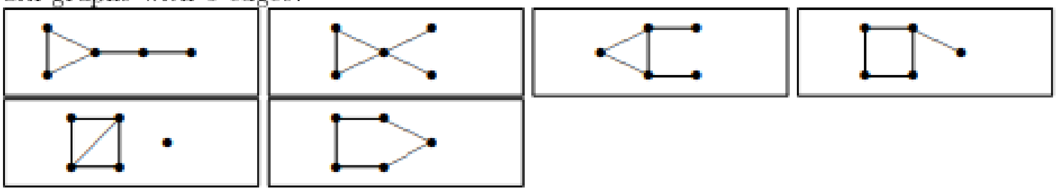
Four graphs with 3 edges:



Six graphs with 4 edges:



Six graphs with 5 edges:



Graphs with 6 (or more) vertices do not need to be considered, since they are obtained from the so-called. a complete graph (i.e. containing all possible edges, in our case 10 edges) by removing the edges corresponding to the ones from already listed graphs. So, for example, there are as many graphs with 2 edges as there are with $10 - 2 = 8$ edges.

By that, the required number is $2 \cdot (1 + 1 + 2 + 4 + 6) + 6 = 34$.

Criteria:

- the given answer is incorrect – 0 points;
- the given answer is correct - 10 points.

Answer: 34

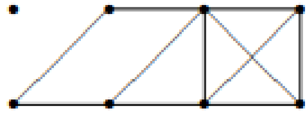
Задача 2. (15 points)

Приведите пример графа G, который содержит все возможные неизоморфные друг другу графы из 4 вершин в качестве вложенных.

Give an example of a graph G that contains all possible non-isomorphic graphs of 4 vertices as its embedded ones.

Решение.

Solution (ENG).



Критерии оценивания:

- нет решения, либо оно неразборчиво – 0 баллов;
- в приведенный граф вложены некоторые графы на 4 вершинах – 5 баллов;
- в приведенный граф вложены почти все (за исключением одного-двух) графы на 4 вершинах – 10 баллов;
- в приведенный граф вложены все графы на 4 вершинах – 15 баллов;

Criteria:

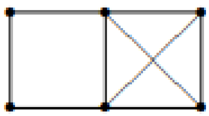
- there is no solution or it is illegible – 0 points;
- the given graph contains some graphs with 4 vertices – 5 points;
- almost all (with the exception of one or two) graphs on 4 vertices are embedded in the given graph – 10 points;
- all graphs on 4 vertices are embedded in the given one – 15 points;

Задача 3. (25 points)
Какое наименьшее число вершин может содержать граф G, описанный в предыдущей задаче? Ответ обоснуйте.

What is the smallest number of vertices that the graph G from the previous task can contain? Explain your answer.

Решение. Существуют 11 попарно неизоморфных графов на 4-х вершинах (см. решение задачи 1 для 4-х вершин и максимального количества рёбер, равного $C_4^2 = 6$). Пример графа G на 8 вершинах приведен в решении задачи 2; осталось доказать, что никакой граф на 7 вершинах не удовлетворяет условию (очевидно, если ему удовлетворяет какой-то граф с $k < 7$ вершинами, то добавлением к нему $(7 - k)$ вершин без рёбер также получим требуемый граф).

Сначала заметим, что полный граф на 4-х вершинах должен быть подграфом G, при этом в него не вложен цикл из 4 вершин, поэтому необходимо добавить еще не менее 2-х вершин к уже имеющимся:



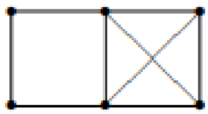
Чтобы получить «пустой» (не содержащий рёбер) граф из изображенного выше, потребуется добавить не менее 2-х вершин, что делает общее число вершин равным 8.
Если бы цикл длины 4 имел всего одну общую вершину с полным графом на 4-х вершинах, то вместе они имели бы 7 вершин, и для вложенности пустого графа понадобилось бы добавить еще как минимум 1 вершину. Доказательство завершено.

Критерии оценивания:

- нет решения (в т.ч. если приведен только ответ) – 0 баллов;
- даны ответ и пример – 10 баллов;
- приведена верная оценка – 15 баллов;
- приведены верные оценка и пример – 25 баллов.

Solution (ENG). There are 11 pairwise non-isomorphic graphs on 4 vertices (see the solution of task 1 for 4 vertices and the maximum number of edges equal to $\binom{4}{2} = 6$). An example of the required graph G on 8 vertices is given in the solution of task 2; it remains to prove that no graph on 7 vertices satisfies the condition (obviously, if some graph with $k < 7$ vertices satisfies it, then by adding $(7 - k)$ vertices without edges to it we will also obtain the required graph).

First, note that a complete graph on 4 vertices must be a subgraph of G, and it does not contain a cycle of 4 vertices as its embedded one, so it is necessary to add at least 2 more vertices to the existing ones:



To get an «empty» (i.e. with no edges) graph from the one shown above, you will need to add at least 2 vertices, which makes the total number of vertices equal to 8.
If a cycle of length 4 had only one common vertex with a complete graph of 4 vertices, then together they would have 7 vertices, and it would be necessary to add at least 1 more vertex to get the empty graph as embedded one. The proof is complete.

Criteria:

- there is no solution (or if only the answer is given) – 0 points;
- the answer and example are given – 10 points;
- the correct estimation is given – 15 points;
- the correct estimation and example are given – 25 points.

Answer: 8

Задача 4. (50 points)
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 Мб

У Никиты было два одинаковых неориентированных графа из n вершин и m ребер, вершины графов пронумерованы различными целыми числами от 1 до n.

Ильнуру не понравилось, что графы одинаковые, и он перенумеровал номера вершин первого графа произвольным образом. В результате перенумерации, номера вершин первого графа все еще являются различными целыми числами от 1 до n.

Вам даны итоговые два графа. Требуется проверить, могли ли эти графы получиться в результате перенумерации, и перенумеровать вершины первого графа обратно, чтобы получились одинаковые графы.

Input file:
Первая строка содержит два целых числа n и m ($1 \le n \le 8$, $0 \le m \le \frac{n(n-1)}{2}$) — количество вершин и ребер графов соответственно.

Каждая из следующих m строк содержит два целых числа u_i и v_i ($1 \le u_i < v_i \le n$) — ребра первого графа.

Следующие m строк задают второй граф в формате, аналогичном первому графу.

Гарантируется, что у каждого из двух графов все ребра попарно различны.

Output file:

Если графы не могли получиться в результате перенумерации, выведите «NO».

Иначе, в первой строке выведите «YES».

На следующей строке выведите n различных целых чисел p_i от 1 до n, означающих, что вершина с номером i первого графа должна получить номер p_i. В результате такой перенумерации должны получиться одинаковые графы.

Вы можете выводить каждую букву в любом регистре (строчную или заглавную). Например, строки «yEs», «yes», «Yes» и «YES» будут приняты как положительный ответ.

Критерии оценивания: баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
0	0	Тесты из условия		полная
1	6	граф является деревом, степень каждой вершины не превосходит 2		первая ошибка
2	11	в графе нет циклов, степень каждой вершины не превосходит 2	1	первая ошибка
3	16	гарантируется, что можно перенумеровать ровно две вершины первого графа		первая ошибка
4	17	нет	$1-3$	первая ошибка

Example 1:

Input:

```
3 2
1 2
2 3
1 3
2 3
```

Output:

```
YES
1 3 2
```

Example 2:

Input:

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

Output:

NO

В первом примере нужно переименовать вершину с номером 2 в 3, и вершину с номером 3 в 2, чтобы получить одинаковые графы.

Во втором примере первый граф — цикл длины 4, а второй — цикл длины 3, к которому подвешена одна вершина, поэтому нельзя получить второй граф из первого перенумерацией вершин.

Time limit: 1 second
Memory usage limit: 256 MB

Nikita had two identical undirected graphs of n vertices and m edges; the vertices of the graphs were numbered by different integers from 1 to n.

Ilnur didn't like that the graphs were the same, and he renumbered the numbers of the vertices of the first graph in an arbitrary way. As a result of renumbering, the vertex numbers of the first graph are still distinct integers from 1 to n.

You are given the final two columns. It is required to check whether these graphs could have been obtained as a result of renumbering, and to renumber the vertices of the first graph back so that identical graphs are obtained.

Input file:

The first line contains two integers n and m ($1 \le n \le 8$, $0 \le m \le \frac{n(n-1)}{2}$) — number of vertices and graph edges, respectively.

Each of the next m lines contains two integers u_i and v_i ($1 \le u_i < v_i \le n$) — edges of the first graph.

The next m lines define the second graph in a format similar to the first graph.

It is guaranteed that each of the two graphs has all edges that are pairwise distinct.

Output file:

If the graphs could not be obtained as a result of renumbering, output «NO».

Otherwise, print «YES» on the first line.

On the next line print n different integers p_i from 1 to n, meaning that vertex number i of the first graph should be numbered p_i. As a result of such renumbering, identical graphs should be obtained.

You can display each letter in any case (lowercase or uppercase). For example, the strings «yEs», «yes», «Yes» and «YES» will be accepted as a positive response.

Criteria: points for each subtask are awarded only if all tests for that subtask and the required subtasks are successfully passed.

Subtask	Points	Additional restrictions	Necessary subtasks	Information about checking
0	0	Tests from condition		complete
1	6	the graph is a tree, the degree of each vertex does not exceed 2		first error
2	11	there are no cycles in the graph, the degree of each vertex does not exceed 2	1	first error
3	16	it is guaranteed that exactly two vertices of the first graph can be renumbered		first error
4	17	no	1 — 3	first error

Example 1:

Input:

3 2
1 2
2 3
1 3
2 3

Output:

YES
1 3 2

Example 2:

Input:

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

Output:

NO

In the first example, you need to rename vertex number 2 to 3, and vertex number 3 to 2 to get identical graphs.

In the second example, the first graph is a cycle of length 4, and the second is a cycle of length 3, to which one vertex is suspended, so it is impossible to obtain the second graph from the first by renumbering the vertices.

Усталый аналитик (10-11 классы) / Tired analyst (10-11 grades)

Фиксирована прямоугольная система координат. Будем называть точку с координатами (x, y) **целой**, если $x, y \in \mathbb{Z}$.

В некоторых целых точках стоят маячки, а в некоторых – датчики (в каждой целой точке – не более одного устройства любого типа), и за всем этим со стороны наблюдает аналитик. Каждый маячок каждую минуту сообщает всем датчикам плоскости свои координаты. Датчики, в свою очередь, способны принять координаты всех маячков и знают свои координаты в любой момент времени.
Каждую минуту каждый датчик «голосует», т.е. сообщает аналитику координаты ближайшего к себе маячка (если таких маячков несколько, датчик сообщает координаты одного из них, выбранного случайно).

В каждый датчик встроены следующие функции вычисления расстояния между точками $A_1(x_1, y_1)$ и $A_2(x_2, y_2)$:

- евклидова: $d_2(A_1A_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$;
- супер-евклидова: $d_n(A_1A_2) = \sqrt[n]{|x_1 - x_2|^n + |y_1 - y_2|^n}$ для заданного аналитиком натурального n;
- манхэттенская: $d_1(A_1A_2) = |x_1 - x_2| + |y_1 - y_2|$;
- равномерная: $d_\infty(A_1A_2) = \max\{|x_1 - x_2|, |y_1 - y_2|\}$.

По умолчанию используется евклидова функция, но аналитик может переключать этот параметр у всех датчиков одновременно.

Аналитик, будучи уставшим, не может следить за сообщениями от всех n датчиков, а каждую минуту выбирает из них набор из k штук и записывает координаты, за которые «проголосовало» большинство датчиков из этого набора (при равенстве голосов аналитик выбирает любые координаты, набравшие наибольшее число «голосов») – так аналитик получает точку P; из полученных таким образом точек строится последовательность $P_1, P_2, \ldots$. Стараясь быть беспристрастным, уставший аналитик каждую минуту выбирает набор из k датчиков, который имеет как можно меньшее пересечение с предыдущим набором, при этом полностью повторять наборы запрещено.

The cartesian coordinate system is fixed. We will call a point with coordinates (x, y) **integer point** if $x, y \in \mathbb{Z}$.

Some integer points have beacons, and some have sensors (at each whole point there is no more than one device of any type), and an analyst watches all this from the outside. Each beacon reports its coordinates to all sensors every minute. The sensors are able to receive the coordinates of all beacons and know their coordinates at any time.
Every minute each sensor «votes», i.e. informs the analyst of the coordinates of the beacon closest to itself (if there are several such beacons, then the sensor reports the coordinates of one of them, chosen at random).

Each sensor has the following functions for calculating the distance between points $A_1(x_1, y_1)$ and $A_2(x_2, y_2)$:

- euclidean: $d_2(A_1A_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$;
- «super-euclidean»: $d_n(A_1A_2) = \sqrt[n]{|x_1 - x_2|^n + |y_1 - y_2|^n}$ for a positive integer n determined by the analyst;

- manhattan: $d_1(A_1A_2) = |x_1 - x_2| + |y_1 - y_2|$;
- uniform: $d_\infty(A_1A_2) = \max\{|x_1 - x_2|, |y_1 - y_2|\}$.

By default, the Euclidean function is used, but the analyst can switch this parameter for all sensors at once.

The analyst, while being tired, cannot monitor messages from all n sensors, but every minute he selects a set of k of them and writes down the coordinates for which the majority of sensors from this set «voted» (if the votes are equal, the analyst selects any coordinates that received the largest number of «votes») – this is how the analyst gets point P; from the points obtained by this way, the sequence $P_1, P_2, \ldots$ is constructed. Trying to be impartial, a tired analyst every minute selects a set of k sensors that has as little overlap as possible with the previous set, and complete repetition of sets is prohibited.

Задача 1. (10 points)
Какое наибольшее число минут усталый аналитик сможет делать свою работу? На плоскости расположены 12 датчиков, из которых аналитик каждую минуту выбирает 7.

What is the greatest number of minutes that a tired analyst can do his job? There are 12 sensors on the plane, from which the analyst selects 7 every minute.

Решение. Количество способов выбрать k датчиков из n равно $C_n^k = \frac{n!}{(n-k)!k!}$. При $n = 12$ и $k = 7$ получим $C_{12}^7 = 792$.

Критерии оценивания:

- ответ неверный – 0 баллов;
- ответ верный – 10 баллов.

Solution (ENG). There are $\binom{n}{k}$ ways to choose k sensors from n given. Thus, the required number is $\binom{12}{7} = 792$.

Criteria:

- the given answer is incorrect – 0 points;
- the given answer is correct - 10 points.

Answer: 792

Задача 2. (20 points)
Две функции расстояния f_1, f_2 называются эквивалентными, если существуют $a, b > 0$: $a \cdot f_1(A_1A_2) \leq f_2(A_1A_2) \leq b \cdot f_1(A_1A_2)$ для любых точек A_1, A_2.

Очевидно, так называемая «супер-евклидова» функция расстояния, встроенная в датчики, при $n = 1$ совпадает с манхэттенской, а при $n = 2$ – с евклидовой. Для каких пар (m, n) функции d_m, d_n эквивалентны? При каких натуральных n являются эквивалентными функции d_n и d_∞? Для каждой пары эквивалентных функций укажите соответствующие (a, b), при необходимости округлив их до сотых. Ответ обоснуйте.

Two functions (for calculating distances) f_1, f_2 are said to be equivalent if there are some $a, b > 0$ such that $a \cdot f_1(A_1A_2) \leq f_2(A_1A_2) \leq b \cdot f_1(A_1A_2)$ for any points A_1, A_2.

Obviously, the so-called «super-euclidean» distance function coincides with the manhattan function for $n = 1$, and with the euclidean function for $n = 2$. For what pairs of positive integers (m, n) are the functions d_m, d_n equivalent? For what positive integers n are the functions d_n and d_∞ equivalent? For each pair of equivalent functions, determine the corresponding pair (a, b), rounding them to 2 decimal places, if necessary. Explain your answer.

Решение. Сначала сравним d_n и d_∞:

$$d_n = \sqrt[n]{|x_1-x_2|^n + |y_1-y_2|^n} \le \sqrt[n]{2\cdot\left(\max\{|x_1-x_2|,|y_1-y_2|\}\right)^n} = \sqrt[n]{2}\cdot d_\infty$$

При этом

$$d_\infty = \max\{|x_1-x_2|,|y_1-y_2|\} = \sqrt[n]{\left(\max\{|x_1-x_2|,|y_1-y_2|\}\right)^n} \le \sqrt[n]{|x_1-x_2|^n + |y_1-y_2|^n} = d_n$$

Итак, для любого натурального n выполнено

$$d_\infty \le d_n \le \sqrt[n]{2}\cdot d_\infty,$$

т.е. функции d_∞ и d_n эквивалентны для любого натурального n, и соответствующая пара $(a,b) = (1, \sqrt[n]{2})$. Из этого следует, что

$$\frac{d_m}{\sqrt[m]{2}} \le d_\infty \le d_m,$$

откуда

$$\frac{d_m}{\sqrt[m]{2}} \le d_n \le \sqrt[n]{2}\cdot d_m$$

для всех пар (m,n) натуральных чисел, т.е. d_m и d_n всегда эквивалентны, и соответствующая пара $(a,b) = (\frac{1}{\sqrt[m]{2}}, \sqrt[n]{2})$.

Критерии оценивания:

- найдены эквивалентные d_m, d_n – $+2$ балла;
- для найденных эквивалентных d_m, d_n найдены соответствующие пары (a,b) – $+2$ балла;
- для найденных эквивалентных d_m, d_n и соответствующих пар (a,b) приведено верное обоснование – $+6$ баллов;
- найдены эквивалентные d_n, d_∞ – $+2$ балла;
- для найденных эквивалентных d_n, d_∞ найдены соответствующие пары (a,b) – $+2$ балла;
- для найденных эквивалентных d_n, d_∞ и соответствующих пар (a,b) приведено верное обоснование – $+6$ баллов;

Solution (ENG). First, lets compare d_n and d_∞:

$$d_n = \sqrt[n]{|x_1-x_2|^n + |y_1-y_2|^n} \le \sqrt[n]{2\cdot\left(\max\{|x_1-x_2|,|y_1-y_2|\}\right)^n} = \sqrt[n]{2}\cdot d_\infty$$

Also

$$d_\infty = \max\{|x_1-x_2|,|y_1-y_2|\} = \sqrt[n]{\left(\max\{|x_1-x_2|,|y_1-y_2|\}\right)^n} \le \sqrt[n]{|x_1-x_2|^n + |y_1-y_2|^n} = d_n$$

By that, for any positive integer n we have

$$d_\infty \le d_n \le \sqrt[n]{2}\cdot d_\infty,$$

thus the functions d_∞ and d_n are equivalent for any positive integer n, and the corresponding pair $(a,b) = (1, \sqrt[n]{2})$. It follows that

$$\frac{d_m}{\sqrt[m]{2}} \le d_\infty \le d_m,$$

where

$$\frac{d_m}{\sqrt[m]{2}} \le d_n \le \sqrt[n]{2} \cdot d_m$$

for all pairs (m, n) of positive integers, i.e. d_m and d_n are always equivalent, and the corresponding pair $(a, b) = (\frac{1}{\sqrt[m]{2}}, \sqrt[n]{2})$.

Criteria:

- found equivalent d_m, d_n – $+2$ points;
- for the found equivalent d_m, d_n the corresponding pairs (a, b) are found – $+2$ points;
- for the found equivalent d_m, d_n and corresponding pairs (a, b) the correct solution is given – $+6$ points;
- found equivalent d_n, d_∞ – $+2$ points;
- for the found equivalent d_n, d_∞ the corresponding pairs (a, b) were found – $+2$ points;
- for the found equivalent d_n, d_∞ and corresponding pairs (a, b) the correct solution is given – $+6$ points;

Задача 3. (20 points)
На следующий день аналитик был отдохнувшим: t минут подряд размер выбираемого им набора был равен общему количеству датчиков, и условие «не повторять наборы» аналитик, конечно, не соблюдал, и функцию расстояния не менял. Также известно, что в течение этих t минут один из маячков каждую минуту передвигался на 1 либо вверх, либо вниз, либо вправо, либо влево (имея возможность менять направление каждую минуту).

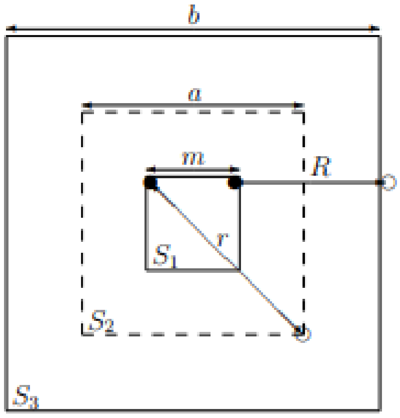
Для каждого натурального t приведите пример расположений маячков, датчиков и последовательности движения одного из маячков, при которых полученная аналитиком последовательность $P_1, P_2, \ldots, P_t$ состояла из попарно различных точек. Для каких t это возможно? Зависит ли ответ от выбора функции расстояния? Объясните свой ответ.

The next day the analyst was rested: for t minutes in a row, the size of the set he selected was equal to the total number of sensors, and the analyst did not comply with the condition «do not repeat sets», also he did not change distance function. It is also known that during these t minutes one of the beacons moved 1 either in right, left, up or down direction (having the ability to change direction every minute).

For each positive integer t, give an example of the locations of beacons, sensors and the sequence of movement of one of the beacons, in which the sequence $P_1, P_2, \ldots, P_t$ obtained by the analyst consisted of pairwise different points. For what t is this possible? Does the answer depend on the choice of distance function? Explain your answer.

Решение. Зафиксируем количество датчиков n и длительность наблюдения t. Пусть m – некоторое натуральное число, не меньшее $\sqrt{n}$; f – выбранная аналитиком функция расстояния.

Расположим все датчики в квадрате S_1 со стороной m и центром в некоторой (не обязательно целой) точке O; подвижный маячок перемещается по периметру квадрата S_2 с тем же центром и стороной $a = \max\{\lceil t/4 \rceil, m+1\}$ ($\lceil x \rceil$ обозначает наименьшее целое число, не меньшее x); все неподвижные маячки расположим снаружи квадрата S_3 с тем же центром O и стороной $b = 2a + 3m + 1$ (стороны всех трёх упомянутых квадратов параллельны координатным осям). Пусть r – наибольшее возможное расстояние от датчика до подвижного маячка, а R – наименьшее расстояние от датчика до неподвижного маячка.



При заданных положениях маячков и датчиков в каждый момент времени ближайшим к любому датчику будет подвижный маячок, координаты которого будут меняться каждую минуту и не повторятся по крайней мере за первые t минут наблюдений – таким образом, последовательность $P_1, P_2, \ldots, P_t$ будет содержать попарно различные точки.

Нетрудно видеть, что показанные на рисунке r и R являются соответственно наибольшим и наименьшим из возможных – значит, осталось доказать, что $R > r$ для любой выбранной f: $R = \frac{b-m}{2}$ и, согласно доказанному в решении задачи 2, $r \leq \sqrt{2} \cdot \frac{a+m}{2}$. Из этого следует, что для $R > r$ требуется $a + m < \frac{b-m}{2}$, т.е. $b > 2a + 3m$, что соответствует выбранным размерам квадратов.

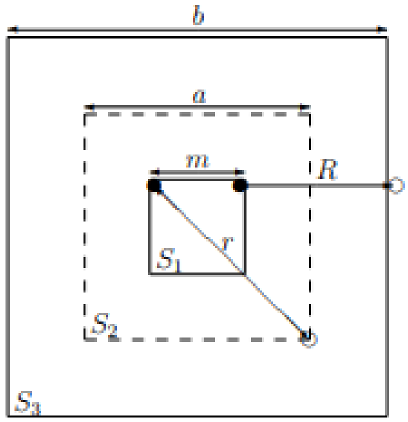
Итак, для любых n и t найдется такая расстановка датчиков и маячков, что точки $P_1, P_2, \ldots, P_t$ будут попарно различны независимо от выбора f.

Критерии оценивания:

- верно найдены требуемые значения t – $+3$ балла;
- дан верный ответ на вопрос про выбор f – $+3$ балла;
- приведена верная стратегия для любого t, но не для любой f – $+5$ баллов;
- приведена верная стратегия для любой f, но не любого t – $+5$ баллов;
- приведена верная стратегия для любых t, f – $+14$ баллов.

Solution (ENG). Lets fix the number of sensors n and the observation time t. Let m be some positive integer not less than $\sqrt{n}$; f is the distance function chosen by the analyst.

Lets place all sensors into a square S_1 with its side m and its center at some (not necessarily integer) point O; a movable beacon moves along the perimeter of a square S_2 with the same center and side $a = \max\{\lceil t/4 \rceil, m+1\}$ ($\lceil x \rceil$ denotes the smallest integer that is not less than x); we place all fixed beacons outside the square S_3 with the same center O and side $b = 2a + 3m + 1$ (the sides of all three mentioned squares are parallel to the coordinate axes). Let r be the greatest possible distance from the sensor to the moving beacon, and R the shortest distance from the sensor to the stationary beacon.



For given positions of beacons and sensors at each moment of time, the closest to any sensor will be a moving beacon, the coordinates of which will change every minute and will not be repeated at least for the first t minutes of observations – thus, the sequence $P_1, P_2, \ldots, P_t$ will contain pairwise distinct points.

It is easy to see that r and R shown in the figure are respectively the largest and smallest possible, which means that it remains to prove that $R > r$ for any chosen f: $R = \frac{b-m}{2}$ and, according to what was proven in task's 2 solution, $r \leq \sqrt{2} \cdot \frac{a+m}{2}$. It follows that $R > r$ requires $a + m < \frac{b-m}{2}$, i.e. $b > 2a + 3m$, which corresponds to the selected squares' sizes.

By that, for any n and t there is a placement of sensors and beacons such that the points $P_1, P_2, \ldots, P_t$ will be pairwise different regardless of the choice of f.

Criteria:

- the required values of t were found correctly – $+3$ points;
- the correct answer to the question about choosing f is given – $+3$ points;
- the correct strategy for any t, but not for any f is given – $+5$ points;
- the correct strategy for any f, but not any t is given – $+5$ points;
- the correct strategy for any t, f is given – $+14$ points.

Задача 4. (50 points)
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 Мб

Даны два натуральных числа: n и k. Рассмотрим множество $\{1, 2, \ldots, n\}$ и все его подмножества размера k. Например, при $n = 3$ и $k = 2$ нас будут интересовать подмножества $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$.

Выпишем все эти подмножества в некотором порядке и рассмотрим пересечения соседних подмножеств. Тогда стоимость расстановки определяется как суммарный размер этих пересечений. От вас требуется найти минимальное возможное значение стоимости расстановки и вывести это значение по модулю $10^9 + 7$.

Input file:
В единственной строке записано два натуральных числа: n ($1 \leq n \leq 10^6$) и k ($1 \leq k \leq n$).

Output file:
Выведите единственное число: минимальное возможное значение стоимости расстановки по модулю $10^9 + 7$.

Критерии оценивания: баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необх. подзадачи
0	0	Тесты из условия	—
1	8	$n \le 4$	—
2	12	$2k+1 \le n$	—
3	17	$n \le 1000$	2
4	13	Нет дополнительных ограничений	1 — 4

Example 1:

Input:
5 1
Output:
0

Example 2:

Input:
4 2
Output:
2

Example 3:

Input:
3 2
Output:
2

Решение. Всего найдется C_n^k k-элементных подмножеств множества $\{1, 2, \ldots, n\}$ всех датчиков ($n > 2$, $k > 0$). Докажем, что минимальная стоимость их расстановки $A_1, A_2, A_3, \ldots, A_{C_n^k}$ равна

- 0, если $k < n/2$: для доказательства построим соответствующую последовательность подмножеств. $A_1 = \{1, 2, \ldots, k\}$, $A_2 = \{k+1, k+2, \ldots, 2k\}$, $A_3 = \{2, 3, \ldots, k, 2k+1\}$, $A_4 = \{1, k+1, k+2, \ldots, 2k-1\}$, $\ldots$ – таким образом будут получены все k-элементные подмножества, при этом любые два последовательные из них не будут иметь пересечений, т.к. у каждого из упомянутых подмножеств $C_{n-k}^k > 1$ возможных «соседей» в последовательности, т.е. последовательность может быть продолжена до тех пор, пока не будут исчерпаны все k-элементные подмножества.

- $\left[\frac{1}{2}(C_n^k - 1)\right]$, если $k = n/2$ ($[t]$ обозначает наибольшее целое число, не превосходящее t): для доказательства построим соответствующую последовательность подмножеств. $A_1 = \{1, 2, \ldots, k\}$, $A_2 = \{k+1, k+2, \ldots, n\}$, $A_3 = \{2, 3, \ldots, k+1\}$, $A_4 = \{1, k+2, k+3, \ldots, n\}$, $\ldots$ – аналогично случаю $k < n/2$, у каждого подмножества может быть $k > 1$ соседей, поэтому начатая последовательность может быть продолжена. При этом каждая вторая пара соседей будет иметь 1 общий элемент, чем и обуславливается найденная минимальная стоимость расстановки.

- $(2k - n)(C_n^k - 1)$, если $k > n/2$: в данном случае каждая пара соседей будет иметь не менее $2k - n$ общих элементов. Требуемая последовательность строится из тех же соображений, что и предыдущие две, и может быть продолжена (до исчерпания всех k-элементных подмножеств), поскольку $C_k^{2k-n} > 1$ для $k < n$ (случай $k = n$ тривиален).

Гамильтонов путь в графе – это путь, проходящий через все вершины по одному разу, т.е последовательность всех вершин графа, в которой каждая следующая соединена ребром с предыдущей. Регулярный граф – это такой граф, в котором степени всех вершин равны. Сильно регулярный граф – это регулярный граф, в котором каждые две смежные (соединенные ребром) вершины имеют равное число соседей, и каждые две несмежные вершины имеют равное число соседей.

Для каждой пары (n, k) можно построить граф, вершины которого – k-элементные подмножества $\{1, 2, \ldots, n\}$, и ребрами соединены те, которые имеют заданное число пересечений (0 для $k < n/2$; 0 или 1 для $k = n/2$; $2k - n$ для $k > n/2$). Легко видеть, что для $k \neq n/2$ полученный граф будет сильно регулярным, причем для всех k граф связный и выполняется теорема Хватала (при $n > 2$):

Если в связном графе на $n \geq 3$ вершинах со степенями $d_1 \leq d_2 \leq \cdots \leq d_n$ для любого m из $d_m \leq m < n/2$ следует $d_{n-m} \geq n - m$, то в таком графе существует гамильтонов цикл (а значит, и гамильтонов путь).

Случай $k = n/2$ можно рассмотреть отдельно: сначала построим регулярный граф, в котором красными рёбрами соединим только пары вершин, которые соответствуют непересекающимся k-элементным подмножествам – тогда граф будет представлять из себя k долей по 2 вершины, соединенные ребром. Теперь соединим синими рёбрами те пары вершин, которые соответствуют парам k-элементных подмножеств, имеющим ровно 1 общий элемент. После этого можно показать, что существует гамильтонов путь, в котором красные и синие рёбра чередуются.

Комментарий: от участников олимпиады не требовалось предъявлять и, тем более, доказывать формулу для минимальной стоимости расстановки.

Time limit: 1 second
Memory usage limit: 256 MB

Given two positive integers: n and k. Consider the set $\{1, 2, \ldots, n\}$ and all its subsets of size k. For example, for $n = 3$ and $k = 2$ we will be interested in the subsets $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$.

Let's write out all these subsets in some order and consider the intersections of neighboring subsets. Then the cost of the arrangement is determined as the total size of these intersections. You are required to find the minimum possible value of the placement cost and output this value modulo $10^9 + 7$.

Input file:

The only line contains two positive integers: n $(1 \leq n \leq 10^6)$ and k $(1 \leq k \leq n)$.

Output file:

Print a single number: the minimum possible value of the placement cost modulo $10^9 + 7$.

Criteria: points for each subtask are awarded only if all tests for that subtask and the required subtasks are successfully passed.

Subtask	Points	Restrictions	Required subtasks
0	0	Tests from condition	–
1	8	$n \leq 4$	–
2	12	$2k + 1 \leq n$	–
3	17	$n \leq 1000$	2
4	13	No additional restrictions	1 – 4

Example 1:

Input:

5 1

Output:

0

Example 2:

Input:

4 2

Output:

2

Example 3:

Input:

3 2

Output:

2

Solution (ENG). There are C_n^k k-element subsets of the set $\{1, 2, \ldots, n\}$ of all sensors ($n > 2$, $k > 0$). Lets prove that the minimum cost of their arrangement $A_1, A_2, A_3, \ldots, A_{C_n^k}$ is equal to

- 0 if $k < n/2$: to prove it, we construct the corresponding sequence of subsets. $A_1 = \{1, 2, \ldots, k\}$, $A_2 = \{k+1, k+2, \ldots, 2k\}$, $A_3 = \{2, 3, \ldots, k, 2k+1\}$, $A_4 = \{1, k+1, k+2, \ldots, 2k-1\}$, … – this will use all k-element subsets, and any two consecutive of them will not have intersections, because each of the mentioned subsets has $C_{n-k}^k > 1$ possible «neighbors» in the sequence, i.e. the sequence can be continued until all k-element subsets have been used.

- $\left[\frac{1}{2}(C_n^k - 1)\right]$ if $k = n/2$ ($[t]$ denotes the largest integer not exceeding t): to prove it, we construct the corresponding sequence of subsets. $A_1 = \{1, 2, \ldots, k\}$, $A_2 = \{k+1, k+2, \ldots, n\}$, $A_3 = \{2, 3, \ldots, k+1\}$, $A_4 = \{1, k+2, k+3, \ldots, n\}$, … – similarly to the case $k < n/2$, each subset can have $k > 1$ neighbors, so the sequence can be continued. In this case, every second pair of neighbors will have 1 common element, which determines the found minimum cost of the arrangement.

- $(2k - n)(C_n^k - 1)$ if $k > n/2$: in this case, each pair of neighbors will have at least $2k - n$ elements in common. The required sequence is constructed from the same considerations as the previous two, and can be continued (until all k-element subsets are exhausted) since $C_k^{2k-n} > 1$ for $k < n$ (case $k = n$ is trivial).

A Hamiltonian path in a graph is a path that passes through all the vertices once, i.e. a sequence of all vertices of the graph in which each next one is connected by an edge to the previous one.
A regular graph is a graph in which the degrees of all vertices are equal. A strongly regular graph is a regular graph in which every two adjacent (edge-connected) vertices have an equal number of neighbors, and every two non-adjacent vertices have an equal number of neighbors.

For each pair (n, k), we can construct a graph whose vertices are k-element subsets of $\{1, 2, \ldots, n\}$, and edges connect those that have a given number of intersections (0 for $k < n/2$; 0 or 1 for $k = n/2$; $2k - n$ for $k > n/2$). It is easy to see that for $k \neq n/2$ the resulting graph will be strongly regular, and for all k the graph is connected and Chvatal's theorem holds (for $n > 2$):
If in a connected graph on $n \geq 3$ vertices with degrees $d_1 \leq d_2 \leq \cdots \leq d_n$ for any m from $d_m \leq m < n/2$ it follows $d_{n-m} \geq n - m$, then such a graph contains a Hamiltonian cycle (and hence a Hamiltonian path).

The case $k = n/2$ can be considered separately: first, we build a regular graph in which we connect with red edges only pairs of vertices that correspond to disjoint k-element subsets – then the graph will consist of k shares of 2 vertices connected by an edge. Now let's connect with blue edges those pairs of vertices that correspond to pairs of k-element subsets that have exactly 1 common element. After this, we can show that there is a Hamiltonian path in which red and blue edges alternate.

Comment: the Olympiad participants were not required to present or prove the formula for the minimum cost of arrangement.