Homework #5

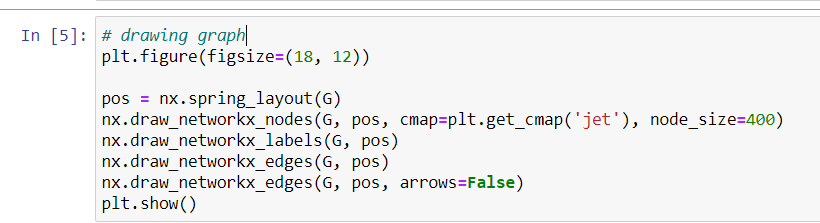
Graph Theory

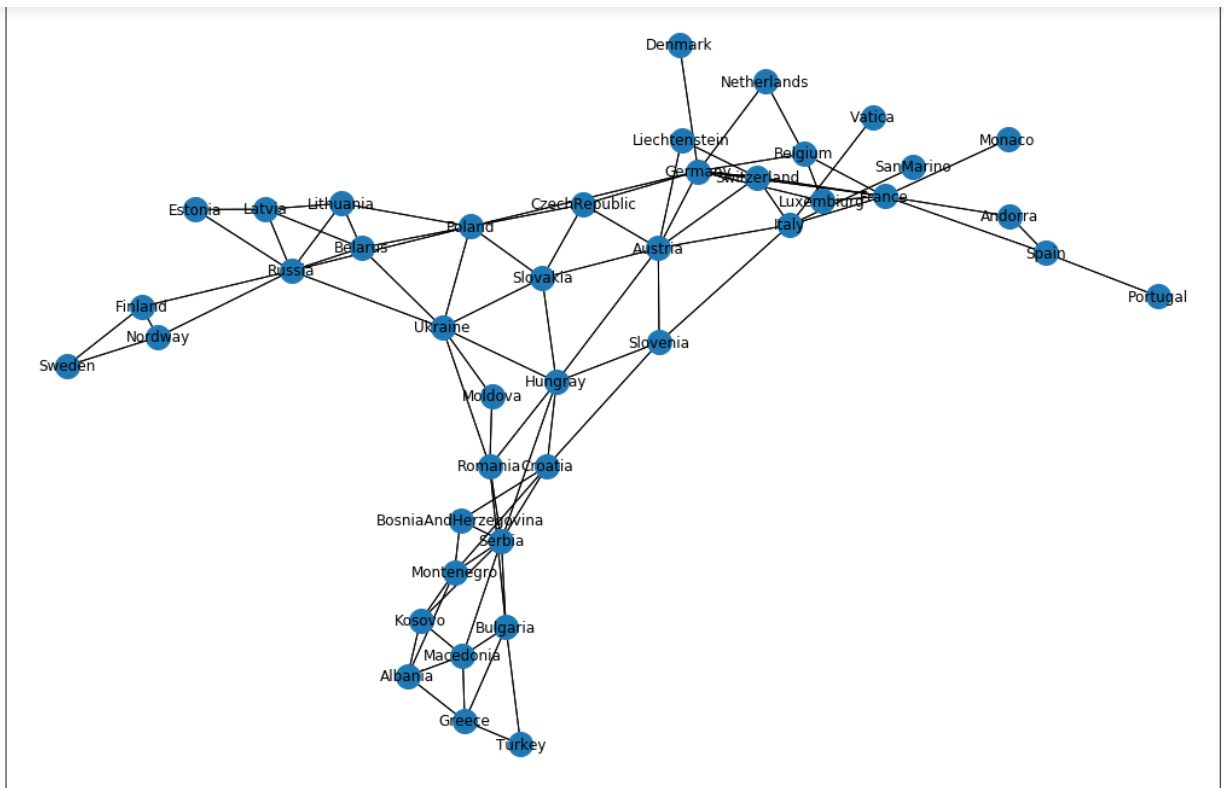
Suslov Misha, M3104

Задание №1.

A : Prove that G is planar by drawing it on a plane without an intersection of edges.

Плоский граф – граф, который можно изобразить на плоскости без пересечения ребёр не по вершинам. То есть, его вершины – точки на плоскости, а рёбра – непересекающиеся кривые на ней.





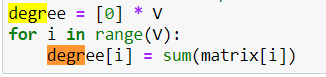
Видно, что изображённых граф удовлетворяет определению плоского графа, т.е. условиям, сл-но является плоским.

B : Find |𝑉 |, |𝐸|, 𝛿 (G), Δ(G), rad(G), diam(G), girth(G), center(G), 𝜅(G), 𝜆(G).

|V| = 42 – число вершин графа

|E| = 87 – кол-во рёбер графа

Степень вершины:

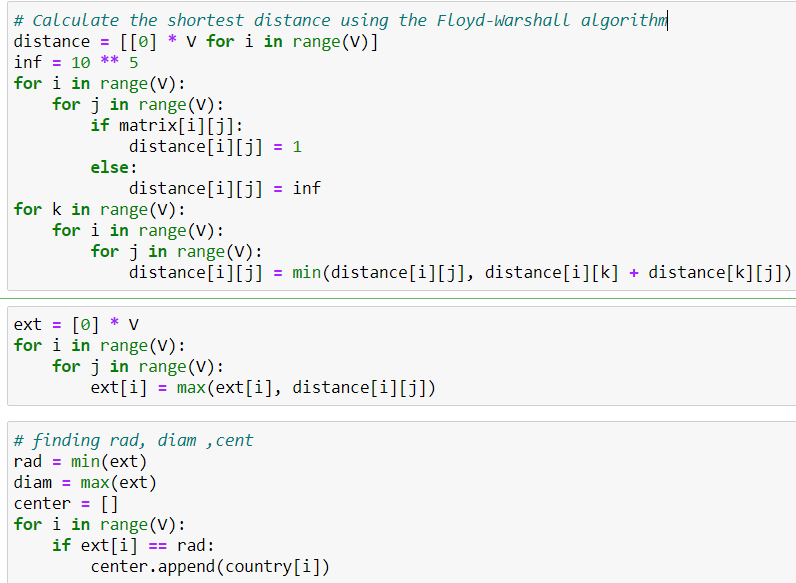


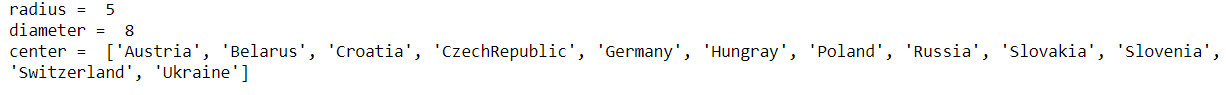




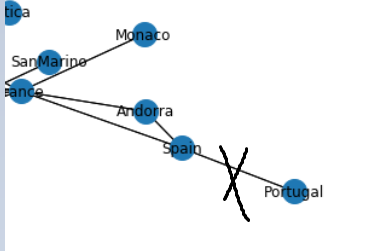
𝛿 (G) = 1

Δ(G) = 9

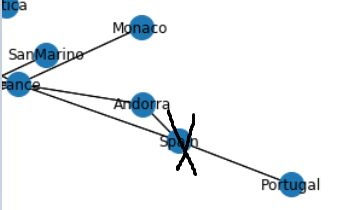
Радиус, диаметр, центр: 



Рёберная связность 𝜆(G):

Убираем ребро между Испанией и Португалией – получаем несвязный граф. 𝜆(G) = 1

Вершинная связность 𝜅(G):

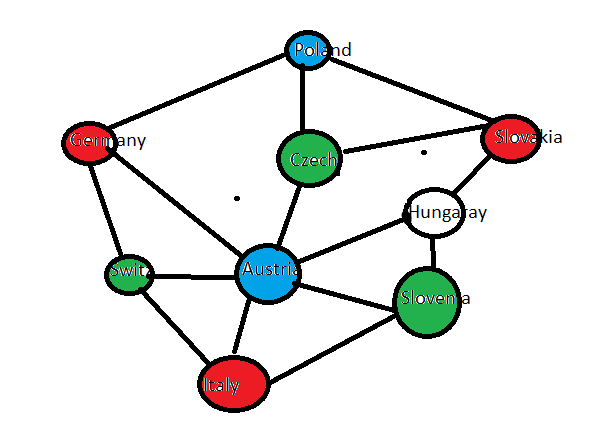
Убираем вершину с Испанией – получаем несвязный граф. 𝜅(G) = 1

C: Find the minimum vertex coloring 𝑍 : 𝑉 →N of G.

Правильная вершинная раскраска – никакие две смежные вершины в ней не получают одного цвета.

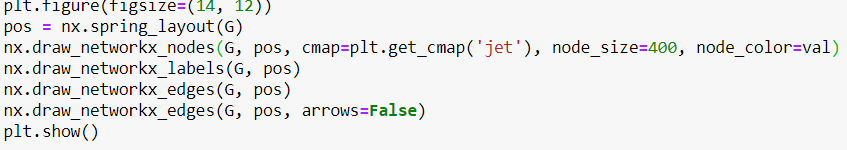
Данный подграф – полный граф, сл-но минимальное кол-во цветов для построения вершинной раскраски больше или равно 3.

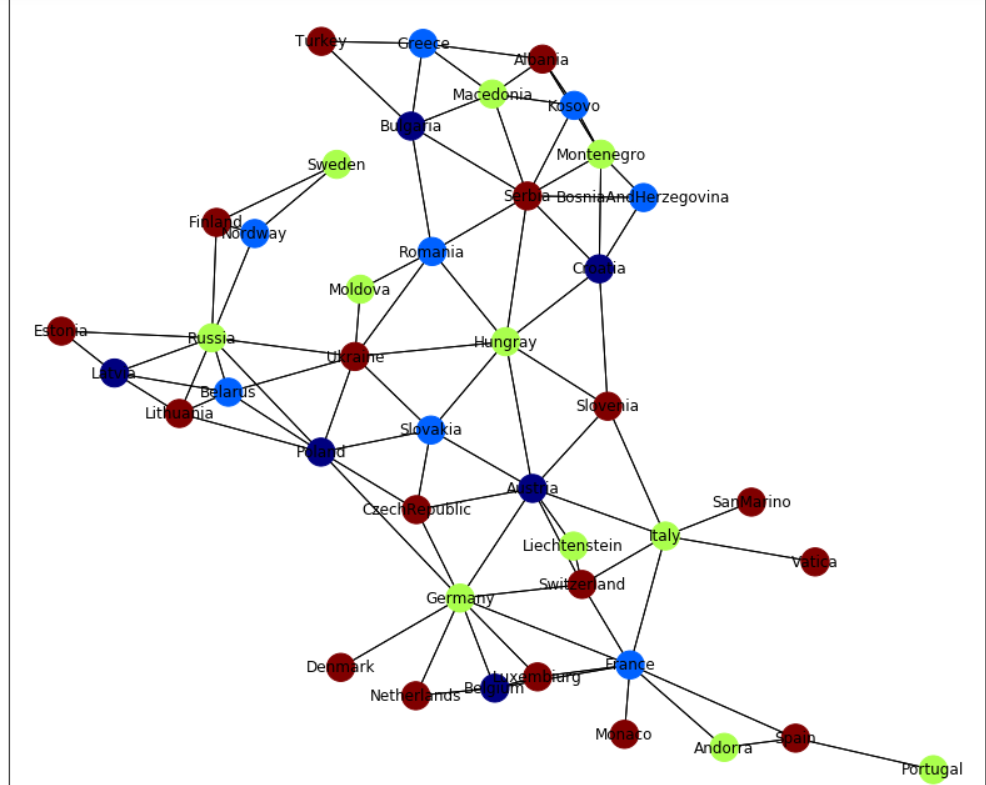
Докажем, что минимальное кол-во цветов будет равно 4:



Пытаемся раскрасить этот граф в три цвета. Рассмотрим вершину Hungary. Она смежна с вершинами, которые имеют три различных цвета : Austria –синий, Slovenia – зелёный и Slovakia – красный. Из этого сделаем вывод, что ей нельзя присвоить ни один из данных трёх цветов. При попытках раскрасить в другие три цвета, результат будет такой же.

С помощью bfs раскрасим граф 4-мя цветами:

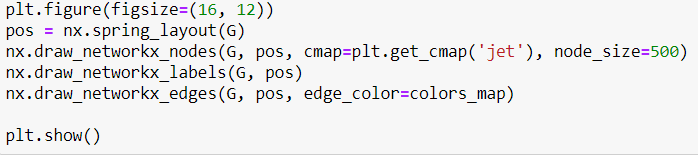


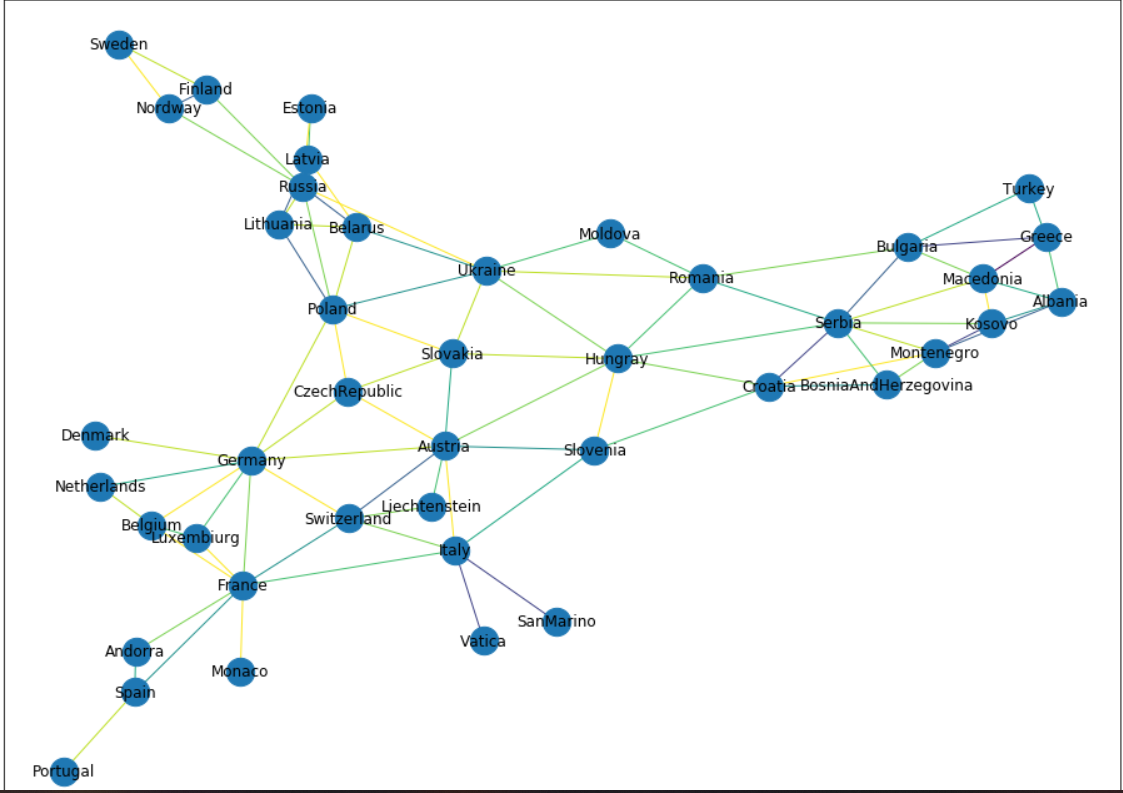


D : Find the minimum edge coloring 𝑋 : 𝐸 →N of G.

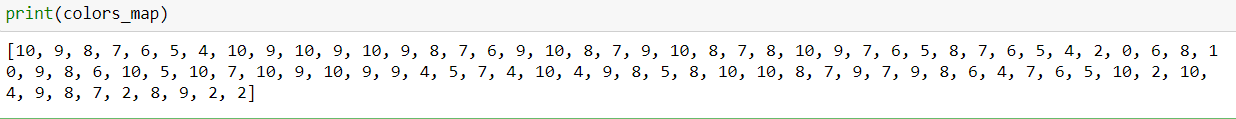
Рёберная раскраска – никакие два смежных ребра не имеют один и тот же цвет.

Вершина “Germany” имеет степень равную 9, сл-но, данная раскраска имеет больше или равно 9 цветов.





Цвета не особо различимы, но вот сам colors\_map:



E: Find the maximum clique 𝑄 ⊆ 𝑉 of G.

Кликой неориентированного графа называется подмножество его вершин, любые две из которых соединены ребром.

Максимальная клика — это клика, которая не может быть расширена путём включения дополнительных смежных вершин, то есть нет клики большего размера, включающей все вершины данной клики. Наибольшая клика — это клика максимального размера для данного графа.



5 клик из 4 вершин. Из условия в “cliques” видно, что нет ни одной клики, где больше четырёх вершин.

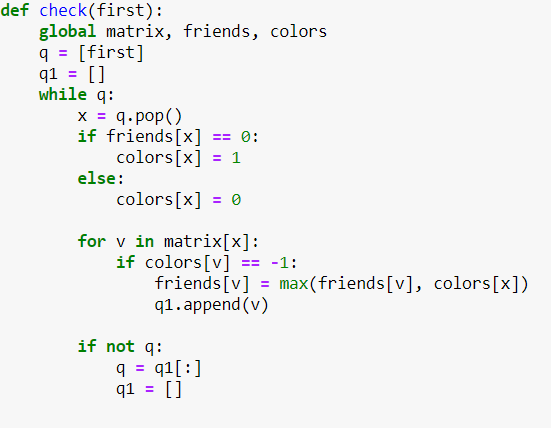
F: Find the maximum stable set 𝑆 ⊆ 𝑉 of G.

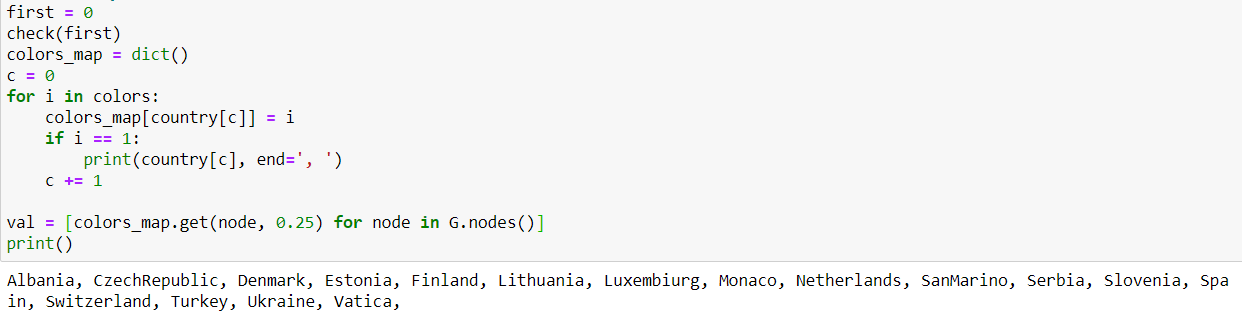
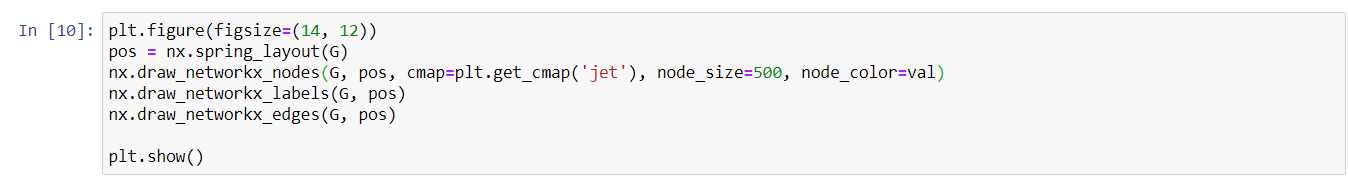
Независимое множество вершин (известное также как внутренне устойчивое множество) есть множество вершин графа G, такое, что любые две вершины в нем не смежны (никакая пара вершин не соединена ребром).

Независимое множество называется максимальным, когда нет другого независимого множества, в которое оно бы входило.

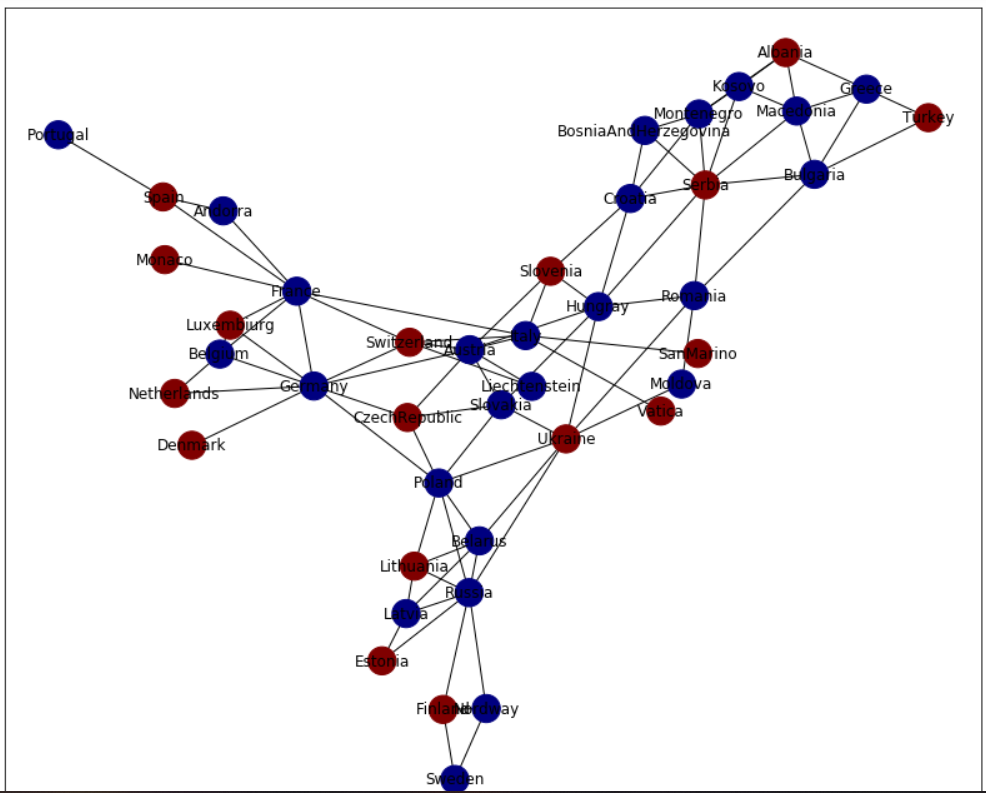
S = {Albania, Czech Republic, Denmark, Estonia, Finland, Lithuania, Luxembourg, Monaco, Netherlands, San Marino, Serbia, Slovenia, Spain, Switzerland, Turkey, Ukraine, Vatican}

Используем BFS.



Вершины из S окрашены в красный цвет:



G: Find the maximum matching 𝑀 ⊆ 𝐸 of G.

Паросочетание в графе - произвольное множество рёбер двудольного графа такое, что никакие два ребра не имеют общей вершины

Максимальное паросочетание - паросочетание, которое не содержится ни в каком другом паросочетании этого графа.

Наибольшее паросочетание – макисимальное паросочетание, содержащее в себе максимальное число рёбер.



H: Find the minimum vertex cover 𝑅 ⊆ 𝑉 of G.

Вершинным покрытием графа G называется такое множество V его вершин, что у любого ребра в G хотя бы одна из вершин лежит в V.

Вершинное покрытие R графа G является минимальных вершинным покрытием графа G, если не существует такого множества U⊆V, что U⊆R

Минимальное вершинное покрытие R графа G является наименьшим вершинным покрытием графа G, если оно содержит минимальное количество вершин.

**Th: Множество независимо тогда и только тогда, когда его дополнение является вершинным покрытием.**

**Consequence: Независимое множество является наибольшим тогда и только тогда, когда его дополнение является наименьшим вершинным покрытием.**

Следовательно, наименьшим вершинным покрытием R является дополнение множества S(maximum stable set):

R = {Albania, Czech Republic, Denmark, Estonia, Finland, Lithuania, Luxembourg, Monaco, Netherlands, San Marino, Serbia, Slovenia, Spain, Switzerland, Turkey, Ukraine, Vatican}

I : Find the minimum edge cover 𝐹 ⊆ 𝐸 of G.

Рёберное покрытие графа — это множество рёбер F, такое, что каждая вершина графа инцидентна по меньшей мере одному ребру из F.

Рёберное покрытие F графа G является минимальным реберным покрытием графа G, если не существует такого множества , что

Минимальное рёберное покрытие F графа G является наименьшим реберным покрытием графа G, если оно содержит минимальное количество рёбер.

Алгоритм: найдём наибольшее паросочетание в графеи включим в него рёбра, необходимые для покрытия непокрытых паросочетанием вершин.

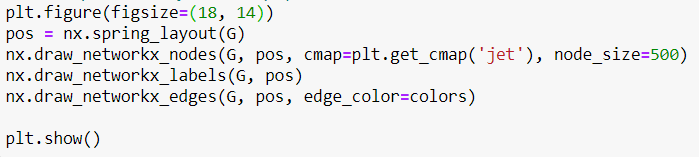
В пункте (g)(Maximum matching) мы нашли наибольшее паросочетание, но оно не покрыло вершины : Denmark, Estonia, Finland, Kosovo, Luxembourg, Monaco, Netherlands, San Marino, Turkey, Vatican

Добавим рёбра множеству F:

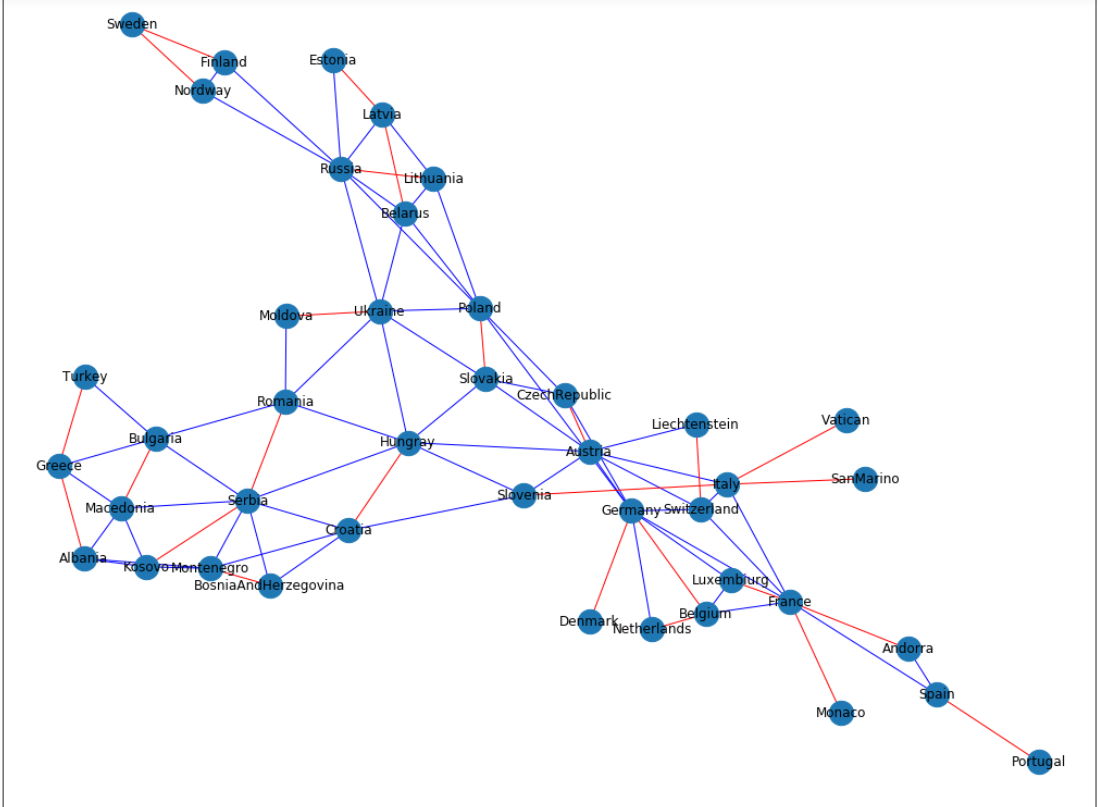
Denmark – Germany, Estonia – Latvia, Finland – Sweden, Kosovo – Serbia, Luxembourg – France, Monaco – France, Netherlands – Belgium, San Marino – Italy, Turkey – Greece, Vatican – Italy

Множество F является наименьшим реберным покрытием графа G:

F = Hungary – Croatia, Liechtenstein – Switzerland, Albania – Greece, Slovenia – Italy, Andorra – France, Belarus – Latvia, Norway – Sweden, Ukraine – Moldova, Denmark – Germany, Estonia – Latvia, Finland – Sweden, Kosovo – Serbia, Luxembourg – France, Monaco – France, Netherlands – Belgium, San Marino – Italy, Turkey – Greece, Vatican – Italy, Slovakia – Poland, Lithuania – Russia, Austria – Czech Republic, Spain – Portugal, Germany – Belgium, Macedonia – Bulgaria, Romania – Serbia, Bosnia And Herzegovina – Montenegro



Снизу изображён граф, где рёбра из F окрашены в красный цвет:



J: Find the shortest closed path (circuit)𝑊 that visits every vertex of G

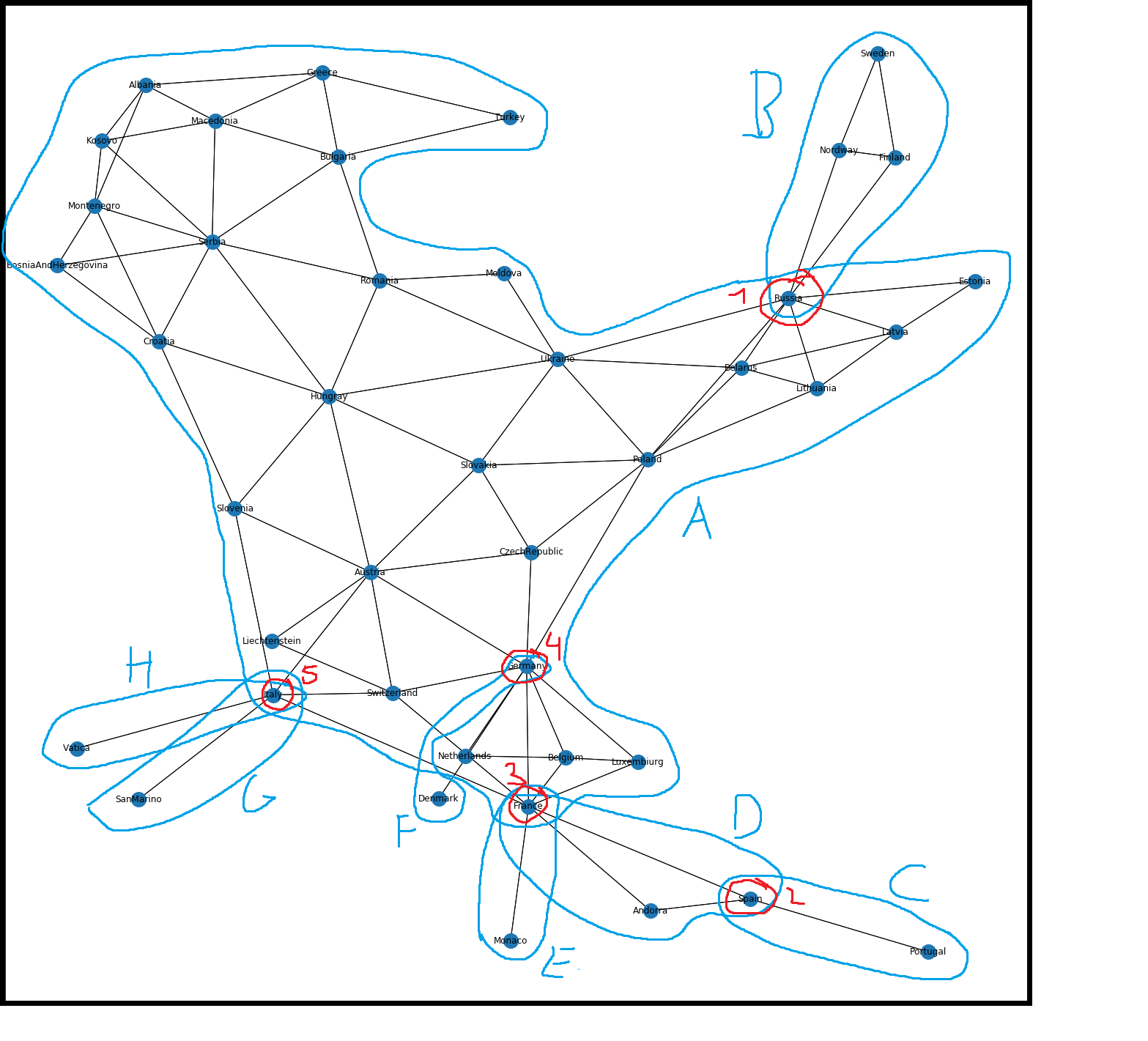
K: Find the shortest closed path (circuit) 𝑈 that visits every edge of G.

Эйлеров путь в графе — это путь, проходящий по всем рёбрам графа и притом только по одному разу.

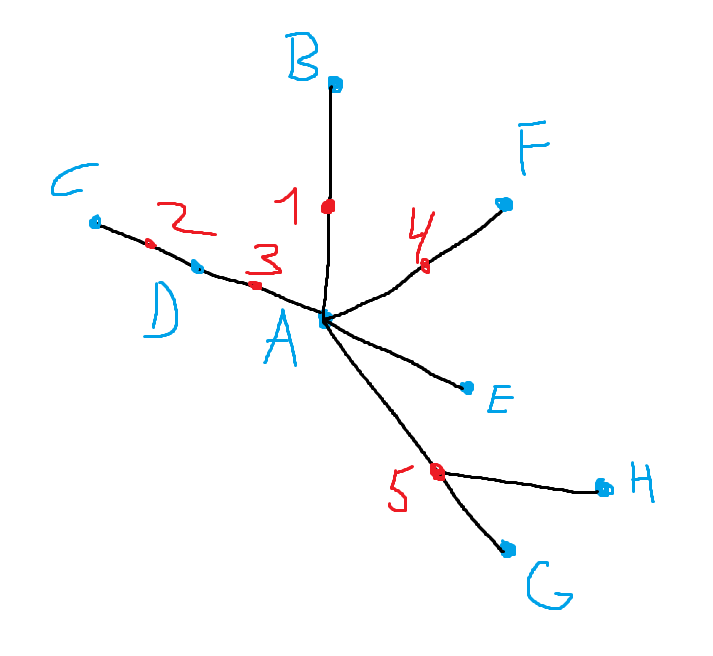
Эйлеров цикл — эйлеров путь, являющийся циклом, то есть замкнутый путь, проходящий через каждое ребро графа ровно по одному разу.



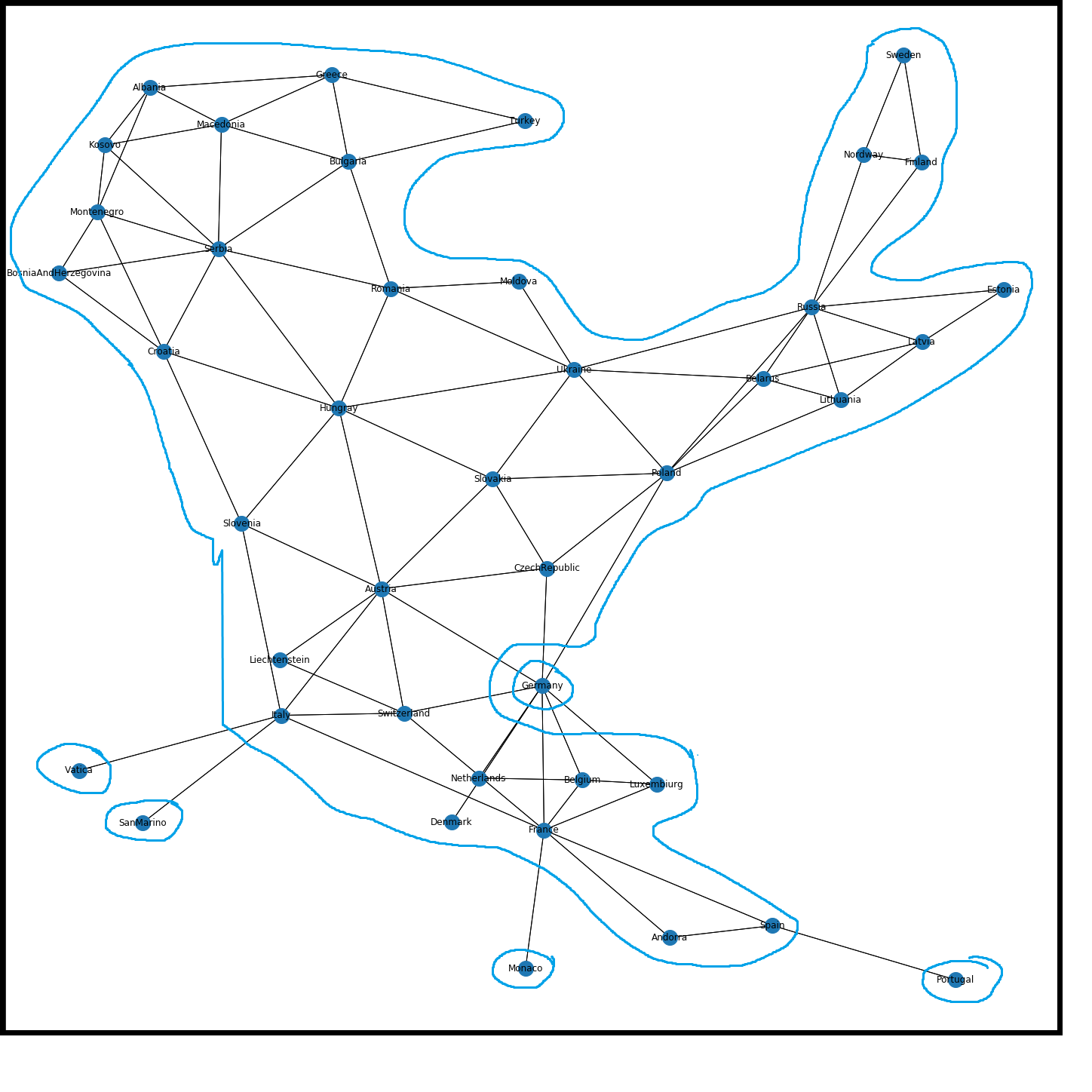
L: Find all 2-vertex-connected components (blocks) and draw a block-cut tree of G∗.



Блоки – синий цвет, шарниры – красный цвет.



M : Find all 2-edge-connected components of G∗.



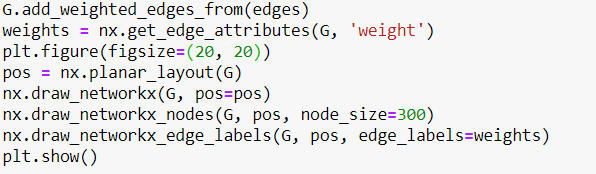
N: Construct an SPQR tree of the largest biconnected component of G.

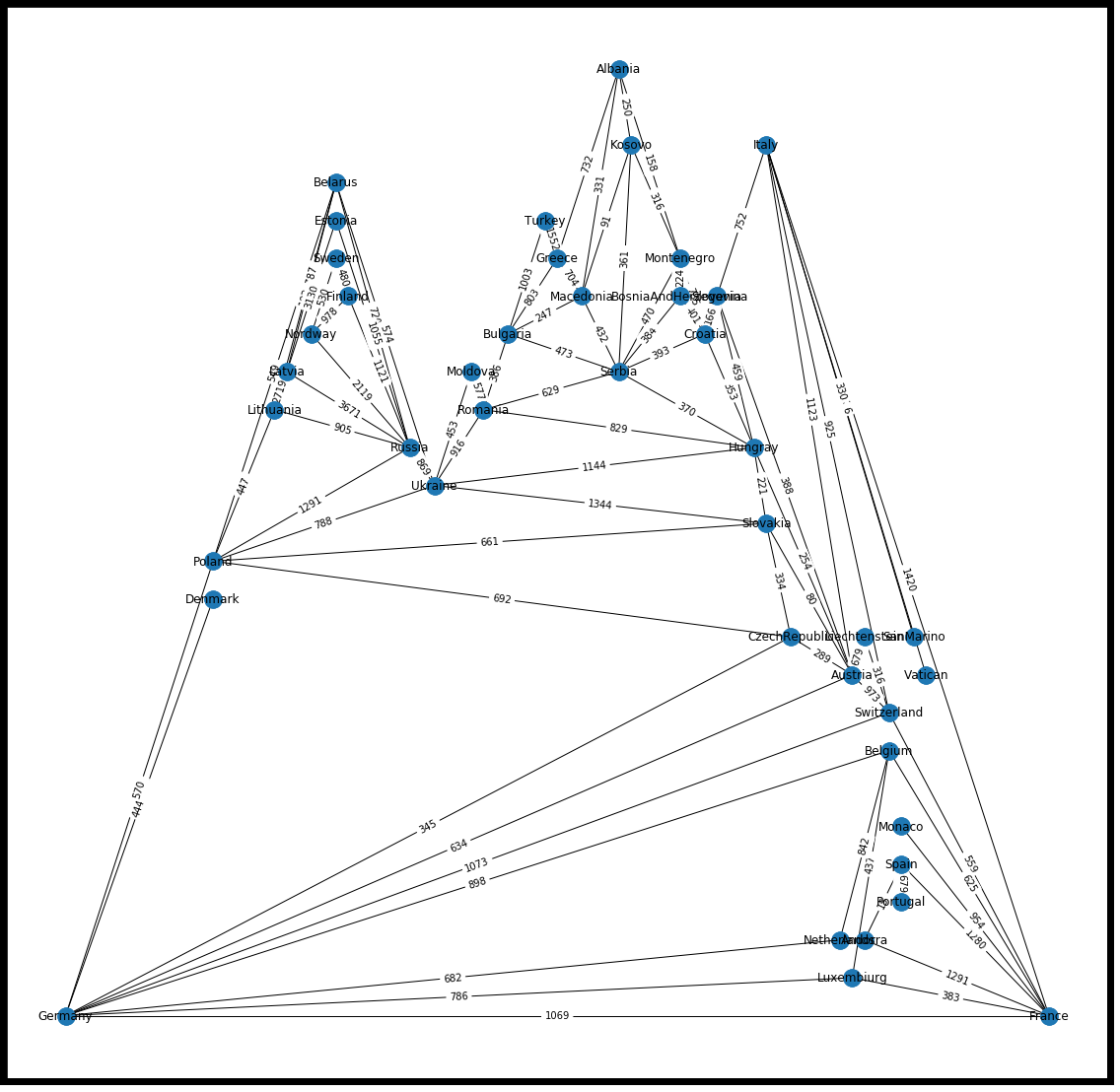
O: Add the weight function𝑤 : 𝐸→R denoting the distance3 between capitals. Find the minimum

(w.r.t. the total weight of edges) spanning tree 𝑇 for the maximum connected component of

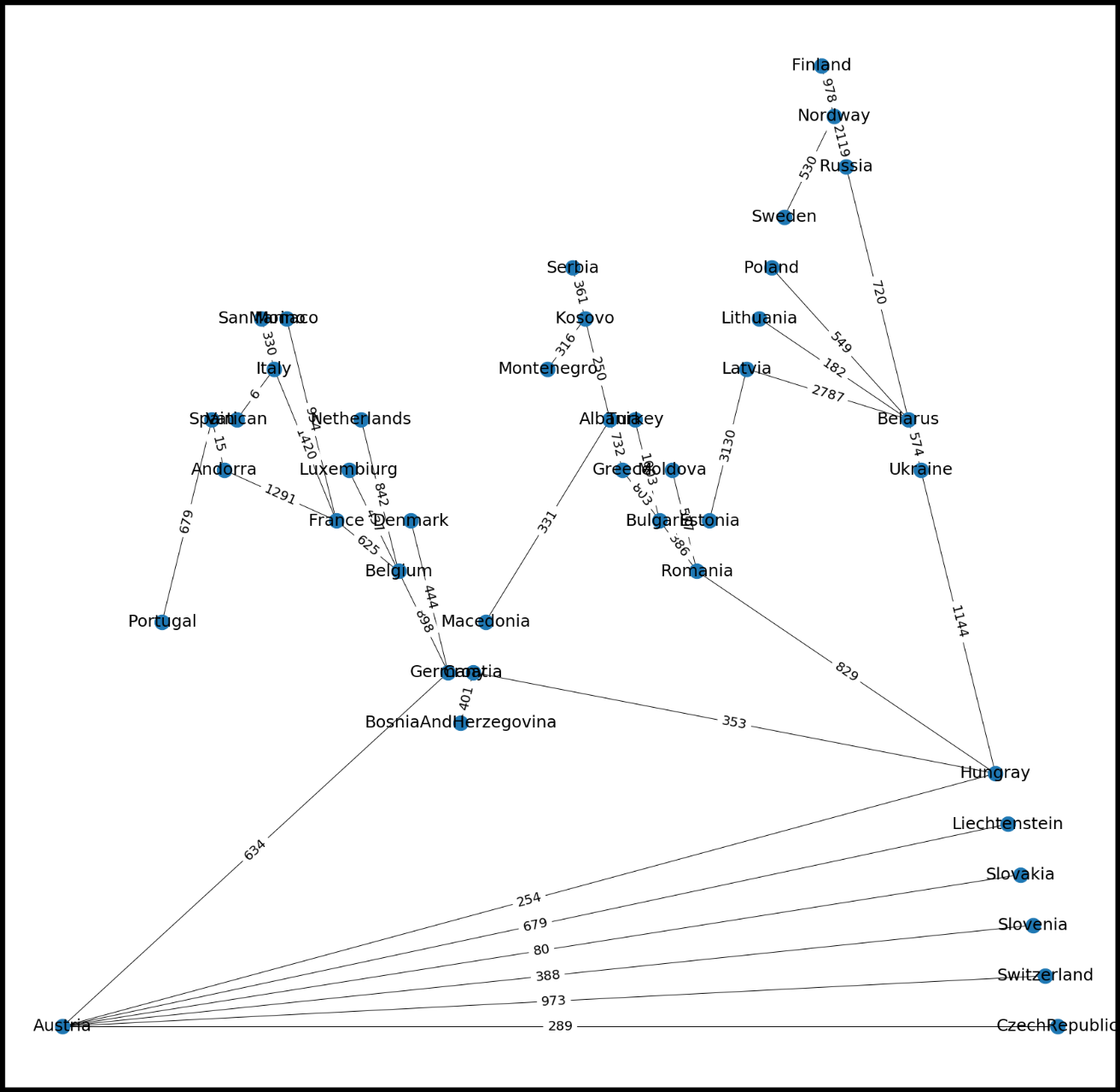
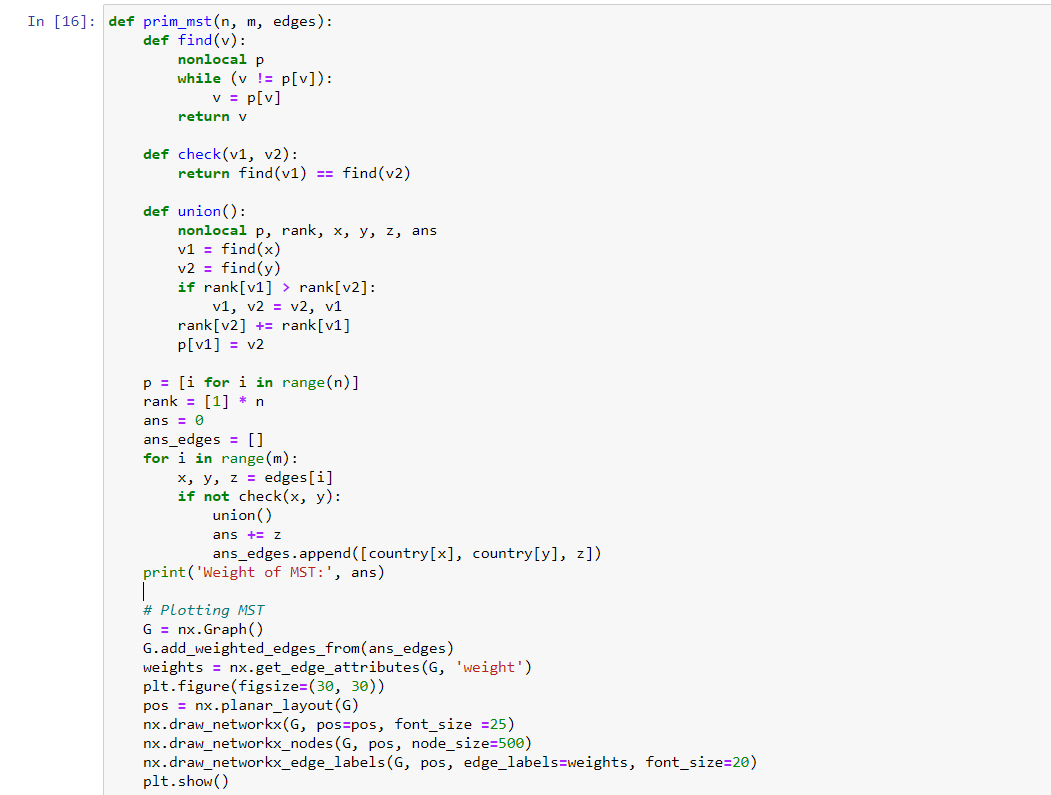
the weighted Europe graph G∗𝑤 = (𝑉, 𝐸,𝑤).

Graph with a weight function:

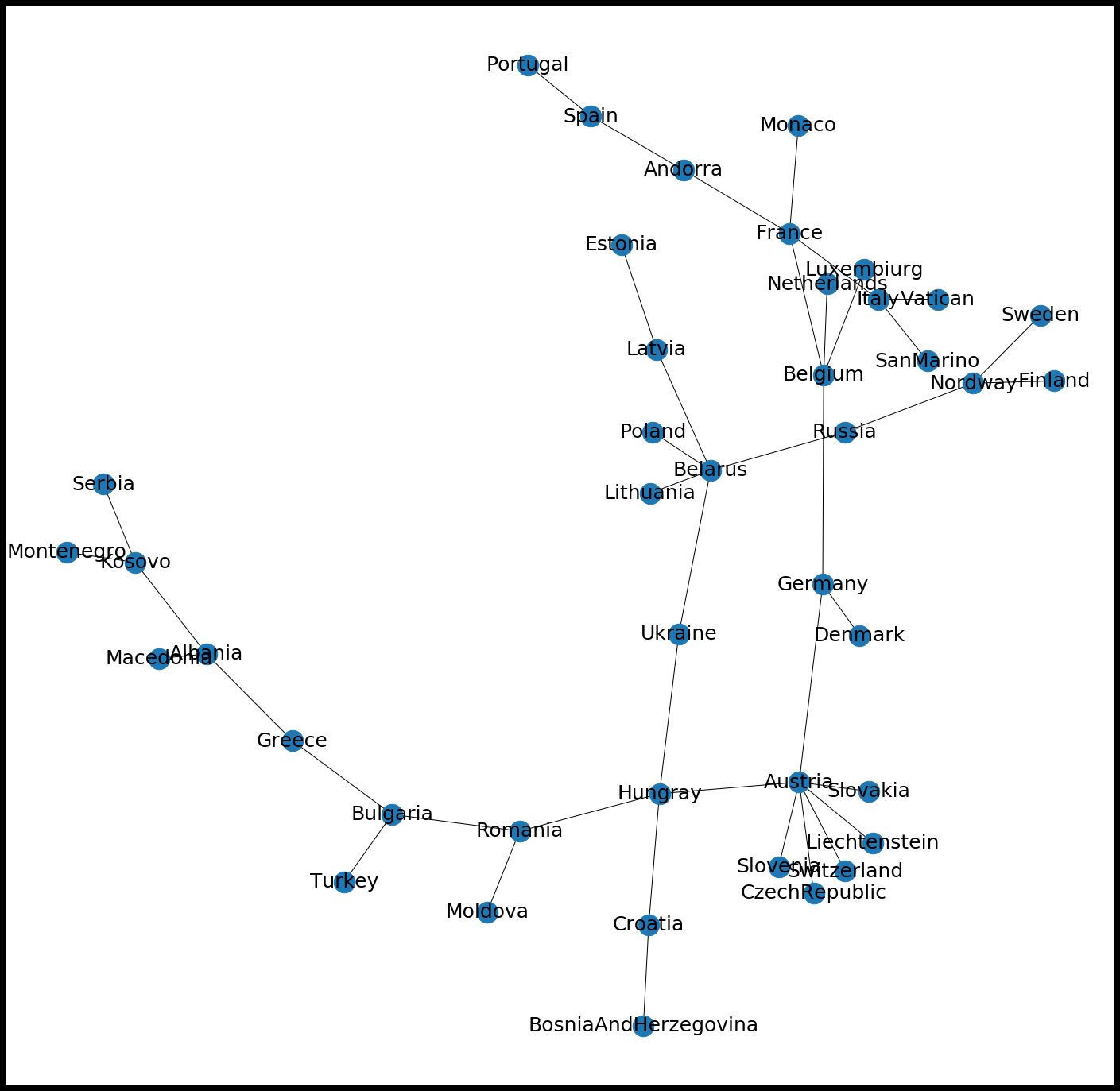


****

Минимальное остовное дерево, полученное с помощью алгоритма Прима:



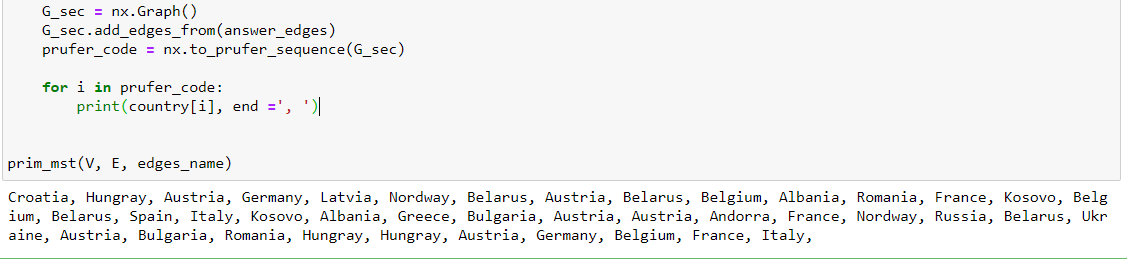
P: Find centroid(𝑇 ) (w.r.t. the edge weight function 𝑤).

Центроидом дерева называется такая вершина v дерева t, после удаления которой дерево разбивается на несколько k поддеревьев 𝑡1, 𝑡2 ,… , 𝑡𝑘, таких что для каждого i: |𝑡𝑖|⩽n/2, то есть размер каждого поддерева не превосходит половины размера исходного дерева.

Центроидом является вершина “Hungary”

Q: Construct the Prufer code for 𝑇 .

Код Прюфера — это способ взаимно однозначного кодирования помеченных деревьев с n вершинами с помощью последовательности n-2 целых чисел в отрезке [1;n]. Иными словами, код Прюфера — это **биекция** между всеми остовными деревьями полного графа и числовыми последовательностями.



Задание №2.

**Theorem 1**. (Triangle Ineqality) For any connected graph 𝐺 = ⟨𝑉, 𝐸⟩:

∀𝑥,𝑦, 𝑧 ∈ 𝑉 : dist(𝑥,𝑦) + dist(𝑦, 𝑧) ≥ dist(𝑥, 𝑧)

Док-во:

Воспользуемся методом “от противного”. Пусть dist(x, y) + dist(y, z) < dist(x, z). Следовательно, сумма длин кратчайших путей между вершинами x и y, и между вершинами y и z меньше, чем длина кратчайшего пути между вершинами x и z. Но dist(x,z) – путь наименьше длины между вершинами x и z. Мы получаем противоречие: пути между двумя вершинами, длина которого меньше длины кратчайшего пути между ними не может быть, сл-но, Triangle Ineqality for any connected graph справедливо.

**Theorem 2.** For any connected graph 𝐺: rad(𝐺) ≤ diam(𝐺) ≤ 2 rad(𝐺).

Док-во:

Т.к. радиус – это минимальный эксцентриситет, а диаметр – максимальный, то очевиден факт, что радиус не может быть больше диаметра.

Пусть u и v такие вершины графа, что dist(u, v) = diam(G). Пусть c – центральная вершина, так что e( c ) = rad(G). Значит, что ни одна вершина не находится на расстоянии большему, чем расстояние, равное радиусу графа, от вершины c. В частности, dist(u, c) и dist(v, c) оба меньше или равны rad(G). Сл-но, dist(u,c)+dist(v,c) ≤2rad(G). По неравенству треугольника, dist(u, v) ≤dist(u,c) + dist(v, c). Из этого следует, что rad(G) ≤diam(G) ≤2rad(G) ч.т.д.