TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

FINAL REPORT DISCRETE STRUCTURES

Final report of Discrete Structures course

Người hướng dẫn: **TS. TRẦN LƯƠNG QUỐC ĐẠI**

Người thực hiện: **PHẠM PHƯỚC TẤN – 520H0418**

Lớp **: 20H50204**

Khoá **: 24**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

FINAL REPORT DISCRETE STRUCTURES

Final report of Discrete Structures course

Người hướng dẫn: **TS. TRẦN LƯƠNG QUỐC ĐẠI**
Người thực hiện: **PHẠM PHƯỚC TẤN**
Lớp : **20H50204**
Khoá : **24**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

LỜI CẢM ƠN

I’m grateful to instruction and explanation of lecturer who is Trần Lương Quốc Đại about the final report of this subject so I could complete effectively my report turning into better.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của TS Trần Lương Quốc Đại. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 07 tháng 01 năm 2022

Tác giả

(ký tên và ghi rõ họ tên)

Phạm Phước Tấn

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

In the final report of this subject so it has 5 problems which need to research including Euclid's algorithm and Bezout's identity, Recurrence relation, Set, Relations, Multiplicative inversion, Kruskal's algorithm, Eulerian circuit, Map coloring.In this problems, I will solve following each step and explaining specifically for my report.

In problem 1 that is Euclid's algorithm and Bezout's identity, I will calculate gcd(a,b), lcm(a,b) and find 5 integer solution pairs (x,y). After solving that, I will be received that is how to calculate gcd, lcm and the equation calculate solution pair (x,y).

In problem 2 that is Recurrence relation, I will be received which is how to calculate characteristic equation and general equation.

In problem 3 that is Set, I will analysis my full name is each word following alphabet order and calculating including union, intersect, non-symmetric difference and symmetric difference.

In problem 4 that is Relations, I will proof that R is reflexive, symetric, anti-symetric, transitive or not ? Based on the valid binary relation to proof it.

In problem 5 that is Multiplicative inversion, I will analysis about Extended Euclidean algorithm to compute multiplicative inverses in modular structures and using it knowledge to calculate multiplicative inverses.

In problem 6 that is Kruskal's algorithm, I will analysis each step to check the circuit of the graph.

In problem 7 that is Eulerian circuit, I will explain about Hierholzer's algorithm to find an Eulerian circuit and define an Eulerian circuit.Using this knowledge to find an Eulerian circuit of the graph.

In problem 8 that is Map coloring, then I will draw again this map to graph and coloring this graph following a minimum number of colors.

MỤC LỤC

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU
CÁC CHỮ VIẾT TẮT

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

DANH MỤC BẢNG

CHAPTER 1 – EUCLID'S ALGORITHM AND BEZOUT'S IDENTITY

1.1 Problem a

My student ID is 520H0418 so m is 418.

I will calculate gcd(2021,1418) and lcm(2021,1418).

The first, I will find gcd(2021,1418):

I will analysis 2021 number which is 1418 number times any number and adding other number be equal 2021 number.I have the general equation:

gcd(a,b) will be a = (a/b) + (a%b) and then I will have the new gcd which is gcd(b,a%b) and analysis until meeting gcd(x,0) then x is the result that I'm looking for this.

I will use algorithms above to solve gcd(2021,1418):

2021 = 1418*1 + 603 =>gcd(1418,603)

1418 = 603*2+212 =>gcd(603,212)

603 = 212*2 + 179 =>gcd(212,179)

212 = 179*1 +33 =>gcd(179,33)

179 = 33*5 +14 =>gcd(33,14)

33 = 14*2 + 5 =>gcd(14,5)

14 = 5*2 + 4 =>gcd(5,4)

5 = 4*1 +1 =>gcd(4,1)

4 = 1*4 +0 => gcd(1,0)

Thus gcd(2021,1418) = 1.

The second, I will find lcm(2021,1418):

I will analysis two number which are 2021 and 1418 number to be the prime factor.

2021 = 43*47 and 1418 = 2*709.

After got the prime factor, I will sort the factors is order from small to large so I will take all of it to times together.

lcm(2021,1418) = 2*43*47*709 = 2865778.

1.2 Problem b

My student ID is 520H0418 so m is 418

I will calculate 2021x + 1418y = gcd(2021,1418). that gcd(2021,1418)=1 is above it which is 1.1 problem a.

I use the result of problem a which is:

gcd(2021,1418)

2021 = 1418*1 + 603 =>gcd(1418,603) =>(-603) = 1418*1-2021 (1)

1418 = 603*2+212 =>gcd(603,212)=>212=1418-603*2 (2)

603 = 212*2 + 179 =>gcd(212,179) =>179 = 603-212*2 (3)

212 = 179*1 +33 =>gcd(179,33) =>33=212-179*1 (4)

179 = 33*5 +14=>gcd(33,14)=> (-14) = 33*5 - 179 (5)

33 = 14*2 + 5 =>gcd(14,5) => 5 = 33-14*2 (6)

14 = 5*2 + 4 =>gcd(5,4) => 4 = 14 -5*2 (7)

5 = 4*1 +1 =>gcd(4,1) (8)

4 = 1*4 +0 => gcd(1,0) = 1.

I have the result from(8) so

5 = 4*1 +1 (I will take 4*1 from the right side to the left side and I will replace from (7) at 4*1)

1 = 5-4*1 = 5 -(14-5*2)*1 (from 7)=5+5*2-14

= 5*3 -14 (I will replace from (6) at 5*3)

=(33-14*2)*3 -14 (from 6)=33*3 -14*6-14

=33*3 - 14*7 (I will replace from (5) at -14*7)

= 33*3 +(33*5 -179)*7 (from 5)=33*3 +33*35 -179*7

= 33*38 -179*7 (I will replace from (4) at 33*38)

=(212-179*1)*38 -179*7 (from 4)=212*38 - 179*38 - 179*7

= 212*38 -179*45 (I will replace from (3) at 179*45)

= 212*38-(603-212*2)*45 (from 3)=212*38 -603*45 + 212*90

= 212*128 -603*45 (I will replace from (2) at 212*128)

=(1418-603*2)*128 - 603*45 (from 2)=1418*128 - 603*256 - 603*45

= 1418*128 - 603*301 (I will replace from (1) at -603*301)

=1418*128+(1418*1-2021)*301(from1)=1418*128+1418*301-2021*301= 1418*429 -2021*301

= 2021*(-301) + 1418*429, So 1 = 2021*(-301) + 1418*429

Based on equation above it which is 2021x + 1418y = 1 so the first solution pair of equation is (x,y) = (-301,429).

I will use the equation which is a.(x+(k.b)/d) + b.(y-(k.a)/d) = d with d = gcd(2021,1418) = 1, a = 2021, b = 1418, x = -301, y = 429.

With k = 2 so I have the result that is 2021(-301 + (2.1418)/1) + 1418.(429-(2.2021)/1) =1 =>2021.2535 +1418.(-3613) = 1. So the second solution pair of equation is (x,y) = (2535,-3613).

With k = 3 so I have the result that is 2021(-301 + (3.1418)/1) + 1418.(429-(3.2021)/1) =1 =>2021.3953 +1418.(-5634) = 1. So the third solution pair of equation is (x,y) = (3953 ,-5634).

With k = 4 so I have the result that is 2021(-301 + (4.1418)/1) + 1418.(429-(4.2021)/1) =1 =>2021.5371+1418.(-7655) = 1. So the fourth solution pair of equation is (x,y) = (5371,-7655).

With k = 5 so I have the result that is 2021(-301 + (5.1418)/1) + 1418.(429-(5.2021)/1) =1 =>2021.6789+1418.(-9676) = 1. So the fifth solution pair of equation is (x,y) = (6789,-9676).

Finally,I will have 5 integer solution pairs (x,y) of this equation: (-301,429), (2535,-3613), (3953 ,-5634), (5371,-7655) and (6789,-9676).

CHAPTER 2 – RECURRENCE RELATION

My student ID is 520H0418 so m is 18

So $a_0 = 5$ and $a_1 = m = 18$

$a_n = 8.a_{n-1} - 15.a_{n-2}$

Characteristic equation:

$t^2 - 8t + 15 = 0 \Rightarrow t_1 = 5, t_2 = 3.$

So t_1 and t_2 is valid value on general equation, so I have:

$a_n = C.5^n + D.3^n$ (*)

With $a_0 = 5$ and $a_1 = 18$ so I will use it on (*):

$$\begin{cases} a_0 = 5 = C.5^0 + D.3^0 = C + D \\ a_1 = 18 = C.5^1 + D.3^1 = 5.C + 3.D \end{cases} \Rightarrow \begin{cases} C = 3/2 \\ D = 7/2 \end{cases}$$

With C=3/2 and D = 7/2 so I will use this value on (*) and I have a_n equation:

$a_n = (3/2).5^n + (7/2).3^n$

CHAPTER 3 – SET

3.1 Problem a

My full name is “Pham Phuoc Tan” so Γ ={A, C, H, M, N, O, P, T, U}

Δ = {A, C, D, G, H, N, O, T, U}.

3.2 Problem b

Γ ={A, C, H, M, N, O, P, T, U}

Δ = {A, C, D, G, H, N, O, T, U}

The union of Γ and Δ which take all element of Γ and Δ following alphabet order but not the same element:

$$\Gamma \cup \Delta = \{A,C,D,G,H,M,N,O,P,T,U\}$$

The intersect of Γ and Δ which just take the elements that Γ and Δ which are the same element and it sorts following alphabet order: (just has “A”, “C” , “H”, “ N”, “O”,”T”, “U” in both Γ and Δ so this is result)

$$\Gamma \cap \Delta = \{A,C,H,N,O,T,U\}$$

The non-symmetric difference of Γ and Δ, it means the elements of Γ that are a same with the elements of Δ so this element isn’t the result. If the elements of Γ is not a same with the elements of Δ then this element is the result: (has “A”, “C”, “H”, “N”, “O”, “T”, “U” of Γ is a same with the elements of Δ,just has “M”, “P” is not a same with the elements of Δ so it’s the result)

$$\Gamma - \Delta = \{M,P\}$$

The symmetric difference of Γ and Δ which just take elements of Γ and Δ that this elements is not the same element of Γ and Δ and this result sort following alphabet order: (has “A”, “C”, “H”, “N”, “O”, “T”, “U” is the elements in both Γ and Δ so it isn’t the result. Just has “M”, “P” of Γ and “D”, “G” of Δ that it’s not a same elements so this is the result)

$$\Gamma \ominus \Delta = \{D,G,M,P\}$$

CHAPTER 4 – RELATIONS

My student ID is 520H0418 so m is 18 then the valid binary relation is $\forall a,b \in N$ ($aRb \leftrightarrow 18|(a.b)$).

Some sets of $\Re$:

$\Re$={(1,18),(18,1),(2,9),(9,2),(3,6),(6,3),(4,9),(9,4),(9,10),(10,9),(6,6),(18,18)}

Reflexive: Suppose, I will take a couple of $\Re$ which is (2,9) with a = 2, b= 9, so $\forall a,b \in N$, $18|(a.a) = 18|(2,2) = 18|4$ but 18 is not divisor of 4 because a.a = d.k so 4 = k.18 and the integer k is not valid value. So $\Re$ isn't reflexive.

Symmetric: Suppose, I will take two couples of $\Re$ including (3,6),(6,3) with a and b following in order (a,b).It's obvious that $\forall a,b \in N$,if 18|(a.b) then 18|(b.a) so it's 18|(3.6) then 18|(6.3). Due to 18|(a.b) so $a.b = k_1.18 \Rightarrow 3.6 = 18.k_1$ so $18=18.k_1 \Rightarrow k_1 = 1$ and 18|(b.a) so $b.a = 18.k_2 \Rightarrow 6.3 = 18.k_2 \Rightarrow 18=18.k_2 \Rightarrow k_2 = 1$.Due to two k_1 and k_2 value in 18|(a.b) and 18|(b.a) is equal and thus aRb -> bRa => so $\Re$ has be Symmetric.

Anti-symmetric: Suppose, I will take a couple of $\Re$ including (6,6) with a and b following in order (a,b).Clearly,if $\forall a,b \in N$,if (aRb^bRa)-> a=b. Due to a = b so aRb^bRa as 6 = 6 so 6R6^6R6. Because of this, a.b = 6.6 =36 so 18|36 is valid value because 36 is multiple of 18 and a.b = 18.k => 36 = 18.k so k = 2 so $\Re$ has be Non-Symmetric.

Transitive: Suppose, I will take two couples of $\Re$ including (2,9),(9,10) with a and b following in order (a,b) which is first couple,then b and c following in order (b,c) which is second couple. Clearly $\forall a,b \in N$, if 18|(a.b) ^ 18|(b.c) then it will be 18|(a.c) but two couples that are (2,9),(9,10) so it's 18|(2.9)^18|(9.10) then it will be 18|(2.10) so 2.10 = 20 so number 18 is not divisor of number 20. So $\Re$ hasn't be Transitive.

CHAPTER 5 – MULTIPLICATIVE INVERSION

5.1 Problem a

Extended Euclidean algorithm to compute multiplicative inverses in modular structures which calculate two integers that are a and m and the algorithm will find modular multiplicative inverse of ‘a’ under modulo ‘m’ and the result of it is positive integer. It will equation is: $x \equiv a^{-1} \pmod{m}$ or $a.x \equiv 1 \pmod{m}$ and x will be from 1 to m-1.

The first is check that calculate the gcd (Greatest Common Divisor) of a and m that check the multiplicative inverse of a modulo m exist or not.It exist if and only if a and m are relatively prime or calculate gcd(a,m)=1.

Step 1: if gcd(a,m) = 1 so I will find the gcd (Greatest Common Divisor) of m and a (it is wrote gcd(m,a)) is showing each step until I get the result which is number 1 of gcd(m,a). If gcd(m,a) is get one so I can move over to step two.

Step 2: Express 1 as the difference between multiples of ‘m’ and ‘a’, use the result of step 1 and express it following number 1. So I will start in the last result of step 1 because it has number 1 and then I will move it to the side and all remaining numbers is moving the different side with number 1.After this, it will transform from down to up is like in the last result is has the number 1.At the same time,it will still replace each other until I meet the modulo m times with other number and the number 1 is still in the fixed side being different with all remaining numbers.The last result of step 2 is the number 1 is still in the fixed side and all remaining numbers are including ‘a’ times to other number and modulo ‘m’ times other number .So I will move over to final step which is step 3.

Step 3: Apply modulo ‘m’ to both side. I will use the last result of step 2 which is the number 1 is still in the fixed side and all remaining numbers are including ‘a’ times to other number and modulo ‘m’ times other number. So I will apply

modulo ‘m’ to both side of equation.Clearly, m times other number is zero because modulo ‘m (mod m)’ is zero, anything times modulo ‘m’ is zero. So modulo ‘m’ times to other number is zero. So the result at present is number 1 modulo ‘m’ and integer ‘a’ times to other number times to modulo ‘m’. and number 1 times modulo ‘m’ it will be number 1 because 1 = (m*0) +1 . So the result at present is number 1 and integer ‘a’ times to other number mod to modulo ‘m’. so I will move the mod modulo ‘m’ to number 1 side so it is the number 1 mod modulo ‘m’ and integer ‘a’ times other number so it is like equation in definition part which is equation is: $x = a^{-1} \pmod m$ or $a.x = 1 \pmod m$ and it’s similar with $a.x = 1 \pmod m$ so x is the result which is a number that times to integer a.Finally, the result of it is the number times to integer a.

5.2 Problem b

My student ID is 520H0418 so m is 18.

I need to find $19^{-1} \pmod{101}$.

The first, I will calculate gcd(19,101), it has the result which is 1 so The multiplicative inverse of 19 module 101 exist because 19 and 101 are relatively prime if gcd(19,101) is 1.

The result of this is a number that ranges from 1 to 100.

Step 1, due to gcd(19,101) = 1 so I will calculate gcd(101,19) to show each step. With the number 101 which is a modulus then number 101 is equal to 19 times by some positive number if I take 101 divide it by 19 then I will find out that 19 goes into 5 times but 19 times 5 is just 95 so I will add 95 to 6 it is equal 101.

$$101 = 19.(5) + 6$$

Continue, I will take 19 is equal to 6 times other number by taking number 19 divide number 6 so the result is number 3 but number 6 times number 3 it will be number 18 so I will plus it to number 1 and then it will be number 19.

$$19 = 6.(3) + 1$$

Due to 6 times 3 and plus 1 so I will stop divide it at here because if continue, then I will take number 6 divide to number 1 so it will also be number 6 so I will stop at here and moving over step 2.

Step 2,Express 1 as the difference between multiples of 101 and 19. From the last result of step 1 so I have 19 = 6.(3) + 1 so I will exchange 6.(3) from the right side to the left side.

$$19 + 6.(-3) = 1 \qquad (1)$$

Continue, So I use the first result of step 1 is 101 = 19.(5) + 6, I will exchange 19.(5) from the right side to the left side.

$$101 + 19.(-5) = 6 \qquad (2)$$

From (1), I will replace (2) on (1):

$$19 + 6.(-3) = 1 \qquad \text{(From (1))}$$

$$19 + [101 + 19.(5)].(-3) = 1$$

$$19 + 101.(-3) + 19.(15) = 1$$

$$101.(-3) + 19.(16) = 1 \qquad (3)$$

Step 3,Apply modulo 101 to both side from the last result of step 2 so it will be:

$$101.(-3) \pmod{101} + 19.(16) \pmod{101} = 1 \pmod{101}$$

Notice 101.(-3) (mod 101), Anything times 101 is 0 because number 101 is modulo and 1(mod 101) which is also number 1 .So (3) equation is going into:

$$101.(-3) \pmod{101} + 19.(16) \pmod{101} = 1 \pmod{101} \quad \text{(From (3))}$$

$$19.(16) \pmod{101} = 1$$

And then, I will move mod 101 from the left side to the right side with number 1:

$$19.(16) = 1 \pmod{101} \qquad \text{(by Theorem 8.4.1 (Epp))}$$

Also, it has other writing type :

$$16 = 19^{-1} \pmod{101} \qquad (4)$$

From 4, it is like equation is: $x = a^{-1} \pmod{m}$ or $a.x = 1 \pmod{m}$ and it's similar with $x = a^{-1} \pmod{m}$ so x is number 16.

So finally, a number 16 is the result because it ranges from 0 to 100.

CHAPTER 6 – KRUSKAL'S ALGORITHM

In kruskal's algorithm, all edges of graph must to have weighted because when it browse, it browse each weighted element of this edge following ascending weighted from the smallest weighted to the largest weighted.In graph, it have n-1 edges(with n is a number of vertices).

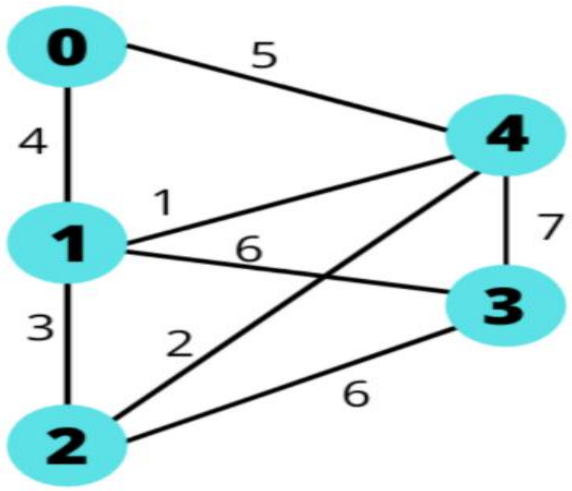
It browse from the smallest weighted to the largest weighted.If it has two weighted be equal then it can occur as output if this edge don't form a cycle.

The important thing in this algorithm is the edges which don't form a cycle.And then, I will remove all parallel edges.

I will sort all the edges in ascending order of weighted.

After sorting this, I start check the edges which is forming a cycle or not. If yes, don't add it on output and don't calculate the weight for minimum cost.If not, I will add it on output and plus this weight of edge with minimum cost.

For instance, I have the graph which is below :



Picture 6: The graph

This graph has 5 vertices and 8 edges.

I will arrange the weighted of the edges following ascending of weighted.

Order Number	Edge Considered	Weighted	Action Taken
1	1 - 4	1	Added
2	2 - 4	2	Added
3	1 - 2	3	Not Added because it form cycle with the edges order number 1 and 2.
4	0 - 1	4	Added
5	0 - 4	5	Not Added because it form cycle with the edges order number 1 and 4.
6	1 - 3	6	Added
7	2 - 3	6	Added
8	3 - 4	7	Not Added because it form cycle with the edges order number 1 and 6 (vertex 1 - 3).

Table 6: Browsing the graph

So finally, the edges in the constructed Graph:

1 - 4 = 1

2 - 4 = 2

0 - 1 = 4

1 - 3 = 6

2 - 3 = 6

Minimum cost of the graph is 1 + 2 + 4 + 6 + 6 = 19.

CHAPTER 7 – EULERIAN CIRCUIT

7.1 Problem a

An Eulerian circuit is circuit which is all vertices have non-zero degree and all vertices have even degree.Also, it starts and ends at the same vertex and a circuit that uses each edge exactly once.

This graph has an Eulerian circuit. Because all vertices have even degree and not any vertex is zero degree.

7.2 Problem b

Hierholzer's algorithm to find an Eulerian circuit which has linear time which is O(E). Choose initial circuit is v vertex and start from v vertex and it will go following a path of edges until returning to v.if the path of any edges which visited so don't go following this path so let start another path which is not visited so it will repeat the previous step until all edges which are visited.

The first: Checking this graph which is undirected graph or not.If the graph has exactly two odd degree vertices then one of this is the starting vertex.Also, the graph don't have any odd degree vertices, then any vertex is the starting vertex.

The second: I will create two stack. The first stack store the final vertices of the Euler Circuit which name is outCircuit.The second stack is contains the waiting vertices which name is TestCircuit.

The third: I will call the starting vertex be v.

Step 1: I will push v in the last TestCircuit.

Step 2: And then I will take the last vertex of TestCircuit to assign it to T which is contains the last vertex of TestCircuit.

Step 3: If all the edges from T are visited, then I will be continue next step and all edges in graph are visited then I want to assign all value of TestCircuit to add in outCircuit.If all edges from T are not visited so let choose any random edge between of T with any X vertex and after this, adding X to TestCircuit.

Step 4: So I will comeback at step 2 and carrying on until the TestCircuit doesn't have the vertex in this.Final, I will assign all the elements of TestCircuit to outCircuit.

7.3 Problem c

My student id is 520H0418 so abcd = 0418, abcd % 4 so 0418%4 = 2 so R1 is UVbaU.

The first: This graph is undirected graph because all vertices is even degree.

The second: I will create two stack. The first stack store the final vertices of the Euler Circuit which name is outCircuit.The second stack is contains the waiting vertices which name is TestCircuit.

The third: I will start vertex be U and the edges include U-V, V-b, b-a, a-U which are visited so I want to add it on the last outCircuit.

After this, I will move over to step 1:

Step 1: I will push U vertex in the last TestCircuit.

Step 2: T = U vertex.(Assign U vertex to T).

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) = (U,T) vertex so I will push T to TestCircuit and the edges between of U and T is visited. The TestCircuit stack includes U,T. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = T vertex.(Assign T vertex to T) because the TestCircuit stack includes U,T.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose Z vertex so random (T,Z) vertex so I will push Z to TestCircuit and the edges between of T and Z is visited. The TestCircuit stack includes U,T,Z. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = Z vertex.(Assign Z vertex to T) because the TestCircuit stack includes U,T,Z.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose a vertex so random (Z,a) vertex so I will push a to TestCircuit and the edges between of Z and a is visited. The TestCircuit stack includes U,T,Z,a. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = a vertex.(Assign a vertex to T) because the TestCircuit stack includes U,T,Z,a.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose g vertex so random (a,g) vertex so I will push g to TestCircuit and the edges between of a and g is visited. The TestCircuit stack includes U,T,Z,a,g. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = g vertex.(Assign g vertex to T) because the TestCircuit stack includes U,T,Z,a,g.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose h vertex so random (g,h) vertex so I will push g to TestCircuit and the edges between of g and h is visited. The TestCircuit stack includes U,T,Z,a,g,h. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = h vertex.(Assign h vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose b vertex so random (h,b) vertex so I will push b to TestCircuit and the edges between of h and b is visited. The TestCircuit stack includes U,T,Z,a,g,h,b. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = b vertex.(Assign b vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose c vertex so random (b,c) vertex so I will push c to TestCircuit and the edges between of b and c is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = c vertex.(Assign c vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose W vertex so random (c,W) vertex so I will push W to TestCircuit and the edges between of c and W is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = W vertex.(Assign W vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose V vertex so random (W,V) vertex so I will push V to TestCircuit and the edges between of W and V is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = V vertex.(Assign V vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose P vertex so random (V,P) vertex so I will push P to TestCircuit and the edges between of V and P is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = P vertex.(Assign P vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose Q vertex so random (P,Q) vertex so I will push Q to TestCircuit and the edges between of P and Q is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = Q vertex.(Assign Q vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose W vertex so random (Q,W) vertex so I will push W to TestCircuit and the edges between of Q and W is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = W vertex.(Assign W vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose X vertex so random (W,X) vertex so I will push X to TestCircuit and the edges between of W and X is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = X vertex.(Assign X vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose d vertex so random (X,d) vertex so I will push d to TestCircuit

and the edges between of X and d is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = d vertex.(Assign d vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose c vertex so random (d,c) vertex so I will push c to TestCircuit and the edges between of d and c is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = c vertex.(Assign c vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose i vertex so random (c,i) vertex so I will push i to TestCircuit and the edges between of c and i is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = i vertex.(Assign i vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,i.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose h vertex so random (i,h) vertex so I will push h to TestCircuit and the edges between of i and h is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = h vertex.(Assign h vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h .
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose m vertex so random (h,m) vertex so I will push m to TestCircuit and the edges between of h and m is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = m vertex.(Assign m vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m .
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose l vertex so random (m,l) vertex so I will push l to TestCircuit and the edges between of m and l is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = l vertex.(Assign l vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose g vertex so random (l,g) vertex so I will push g to TestCircuit and the edges between of l and g is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = g vertex.(Assign g vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose f vertex so random (g,f) vertex so I will push f to TestCircuit and the edges between of g and f is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = f vertex.(Assign f vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose Z vertex so random (f,Z) vertex so I will push Z to TestCircuit and the edges between of f and Z is visited. The TestCircuit stack

includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = Z vertex.(Assign Z vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Z.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose Y vertex so random (Z,Y) vertex so I will push Y to TestCircuit and the edges between of Z and Y is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = Y vertex.(Assign Y vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose S vertex so random (Y,S) vertex so I will push S to TestCircuit and the edges between of Y and S is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = S vertex.(Assign S vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,S.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose T vertex so random (S,T) vertex so I will push T to TestCircuit and the edges between of S and T is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = T vertex.(Assign T vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose N vertex so random (T,N) vertex so I will push N to TestCircuit and the edges between of T and N is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = N vertex.(Assign N vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose M vertex so random (N,M) vertex so I will push M to TestCircuit and the edges between of N and M is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = M vertex.(Assign M vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose S vertex so random (M,S) vertex so I will push S to TestCircuit and the edges between of M and S is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,S.
After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = S vertex.(Assign S vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose B vertex so random (S,B) vertex so I will push B to TestCircuit and the edges between of S and B is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,S,B.
After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = B vertex.(Assign B vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose Y vertex so random (B,Y) vertex so I will push Y to TestCircuit and the edges between of B and Y is visited. The TestCircuit

stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = Y vertex.(Assign Y vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose e vertex so random (Y,e) vertex so I will push e to TestCircuit and the edges between of Y and e is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = e vertex.(Assign e vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose f vertex so random (e,f) vertex so I will push f to TestCircuit and the edges between of e and f is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = f vertex.(Assign f vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose k vertex so random (f,k) vertex so I will push k to TestCircuit and the edges between of f and k is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = k vertex.(Assign k vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose l vertex so random (k,l) vertex so I will push l to TestCircuit and the edges between of k and l is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = l vertex.(Assign l vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k ,l.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose D vertex so random (l,D) vertex so I will push D to TestCircuit and the edges between of l and D is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = D vertex.(Assign D vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose m vertex so random (D,m) vertex so I will push m to TestCircuit and the edges between of D and m is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = m vertex.(Assign m vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose n vertex so random (m,n) vertex so I will push n to TestCircuit and the edges between of m and n is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = n vertex.(Assign n vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose i vertex so random (n,i) vertex so I will push i to TestCircuit and the edges between of n and i is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = i vertex.(Assign i vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose j vertex so random (i,j) vertex so I will push j to TestCircuit and the edges between of i and j is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = j vertex.(Assign j vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose d vertex so random (j,d) vertex so I will push d to TestCircuit

and the edges between of j and d is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = d vertex.(Assign d vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose C vertex so random (d,C) vertex so I will push C to TestCircuit and the edges between of d and C is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = C vertex.(Assign C vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose X vertex so random (C,X) vertex so I will push X to TestCircuit and the edges between of C and X is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = X vertex.(Assign X vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose R vertex so random (X,R) vertex so I will push R to TestCircuit and the edges between of X and R is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = R vertex.(Assign R vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose Q vertex so random (R,Q) vertex so I will push Q to TestCircuit and the edges between of R and Q is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = Q vertex.(Assign Q vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want

to choose L vertex so random (Q,L) vertex so I will push L to TestCircuit and the edges between of Q and L is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = L vertex.(Assign L vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose K vertex so random (L,K) vertex so I will push K to TestCircuit and the edges between of L and K is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = K vertex (Assign K vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose J vertex so random (K,J) vertex so I will push J to TestCircuit and the edges between of K and J is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = J vertex (Assign J vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose I vertex so random (J,I) vertex so I will push I to TestCircuit and the edges between of J and I is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = I vertex (Assign I vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose N vertex so random (I,N) vertex so I will push N to TestCircuit and the edges between of I and N is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = N vertex (Assign N vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose O vertex so random (N,O) vertex so I will push O to TestCircuit and the edges between of N and O is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = O vertex (Assign O vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose J vertex so random (O,J) vertex so I will push J to TestCircuit and the edges between of O and J is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = J vertex (Assign J vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose A vertex so random (J,A) vertex so I will push A to TestCircuit and the edges between of J and A is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = A vertex (Assign A vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose I vertex so random (A,I) vertex so I will push I to TestCircuit and the edges between of A and I is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = I vertex (Assign I vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose E vertex so random (I,E) vertex so I will push E to TestCircuit

and the edges between of I and E is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = E vertex (Assign E vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose M vertex so random (E,M) vertex so I will push M to TestCircuit and the edges between of E and M is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = M vertex (Assign M vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S, B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose B vertex so random (M,B) vertex so I will push B to TestCircuit and the edges between of M and B is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,

S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = B vertex (Assign B vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose e vertex so random (B,e) vertex so I will push e to TestCircuit and the edges between of B and e is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = e vertex (Assign e vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose G vertex so random (e,G) vertex so I will push G to TestCircuit and the edges between of e and G is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = G vertex (Assign G vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose k vertex so random (G,k) vertex so I will push k to TestCircuit and the edges between of G and k is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = k vertex (Assign k vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose D vertex so random (k,D) vertex so I will push D to TestCircuit and the edges between of k and D is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = D vertex (Assign D vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose n vertex so random (D,n) vertex so I will push n to TestCircuit and the edges between of D and n is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n.
After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = n vertex (Assign n vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose H vertex so random (n,H) vertex so I will push H to TestCircuit and the edges between of n and H is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n
,H. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = H vertex (Assign H vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose j vertex so random (H,j) vertex so I will push j to TestCircuit and the edges between of H and j is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = j vertex (Assign j vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose C vertex so random (j,C) vertex so I will push C to TestCircuit and the edges between of j and C is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = C vertex (Assign C vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose R vertex so random (C,R) vertex so I will push R to TestCircuit and the edges between of C and R is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n ,H,j,C,R. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = R vertex (Assign R vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose F vertex so random (R,F) vertex so I will push F to TestCircuit and the edges between of R and F is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n ,H,j,C,R,F. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = F vertex (Assign F vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose L vertex so random (F,L) vertex so I will push L to TestCircuit

and the edges between of F and L is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n ,H,j,C,R,F,L. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = L vertex (Assign L vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose A vertex so random (L,A) vertex so I will push A to TestCircuit and the edges between of L and A is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n ,H,j,C,R,F,L,A. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = A vertex (Assign A vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose E vertex so random (A,E) vertex so I will push E to TestCircuit and the edges between of A and E is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,

S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n
,H,j,C,R,F,L,A,E. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = E vertex (Assign E vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A,E.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose B vertex so random (E,B) vertex so I will push B to TestCircuit and the edges between of E and B is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A,E,B. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = B vertex (Assign B vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A,E,B.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose G vertex so random (B,G) vertex so I will push G to TestCircuit and the edges between of B and G is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,

S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n
,H,j,C,R,F,L,A,E,B,G. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = G vertex (Assign G vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,
l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A,
E,B,G.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose D vertex so random (G,D) vertex so I will push D to TestCircuit and the edges between of G and D is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,
S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n
,H,j,C,R,F,L,A,E,B,G,D. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = D vertex (Assign D vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,
l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A,
E,B,G,D.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose H vertex so random (D,H) vertex so I will push H to TestCircuit and the edges between of D and H is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,

S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n
,H,j,C,R,F,L,A,E,B,G,D,H. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = H vertex (Assign H vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A,E,B,G,D,H.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose C vertex so random (H,C) vertex so I will push C to TestCircuit and the edges between of H and C is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A,E,B,G,D,H,C. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = C vertex (Assign C vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A,E,B,G,D,H,C.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose F vertex so random (C,F) vertex so I will push F to TestCircuit and the edges between of C and F is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,

S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n
,H,j,C,R,F,L,A,E,B,G,D,H,C,F. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = F vertex (Assign F vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,
l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A,
E,B,G,D,H,C,F.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose A vertex so random (F,A) vertex so I will push A to TestCircuit and the edges between of F and A is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,
S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n
,H,j,C,R,F,L,A,E,B,G,D,H,C,F,A. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = A vertex (Assign A vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k,
l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A,
E,B,G,D,H,C,F,A.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose K vertex so random (A,K) vertex so I will push K to TestCircuit and the edges between of A and K is visited. The TestCircuit

stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n ,H,j,C,R,F,L,A,E,B,G,D,H,C,F,A,K. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = K vertex (Assign K vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A, E,B,G,D,H,C,F,A,K.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose P vertex so random (K,P) vertex so I will push P to TestCircuit and the edges between of K and P is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n ,H,j,C,R,F,L,A,E,B,G,D,H,C,F,A,K,P. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = P vertex (Assign P vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A, E,B,G,D,H,C,F,A,K,P.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose O vertex so random (P,O) vertex so I will push O to

TestCircuit and the edges between of P and O is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n ,H,j,C,R,F,L,A,E,B,G,D,H,C,F,A,K,P,O. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = O vertex (Assign O vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A, E,B,G,D,H,C,F,A,K,P,O.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose U vertex so random (O,U) vertex so I will push U to TestCircuit and the edges between of O and U is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n ,H,j,C,R,F,L,A,E,B,G,D,H,C,F,A,K,P,O,U. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = U vertex (Assign U vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A, E,B,G,D,H,C,F,A,K,P,O,U.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want

to choose V vertex so random (U,V) vertex so I will push V to TestCircuit and the edges between of U and V is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n ,H,j,C,R,F,L,A,E,B,G,D,H,C,F,A,K,P,O,U,V. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = V vertex (Assign V vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A, E,B,G,D,H,C,F,A,K,P,O,U,V.
Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose b vertex so random (V,b) vertex so I will push b to TestCircuit and the edges between of V and b is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n ,H,j,C,R,F,L,A,E,B,G,D,H,C,F,A,K,P,O,U,V,b. After this, I will repeat at step 2.

Repeating step 2:
Step 2: T = b vertex (Assign b vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A, E,B,G,D,H,C,F,A,K,P,O,U,V,b.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose a vertex so random (b,a) vertex so I will push a to TestCircuit and the edges between of b and a is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n ,H,j,C,R,F,L,A,E,B,G,D,H,C,F,A,K,P,O,U,V,b,a. After this, I will repeat at step 2.

Repeating step 2:

Step 2: T = a vertex (Assign a vertex to T) because the TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,d,c,I,h,m,l,g,f,Y,T,N,M,S,B,Y,e,f,k, l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A, E,B,G,D,H,C,F,A,K,P,O,U,V,b,a.

Step 3: Due to all edges from T are not visited so let choose any random edge between of T with any X vertex so I will random (T,X) So I want to choose U vertex so random (a,U) vertex so I will push U to TestCircuit and the edges between of a and U is visited. The TestCircuit stack includes U,T,Z,a,g,h,b,c,W,V,P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M, S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n ,H,j,C,R,F,L,A,E,B,G,D,H,C,F,A,K,P,O,U,V,b,a,U.

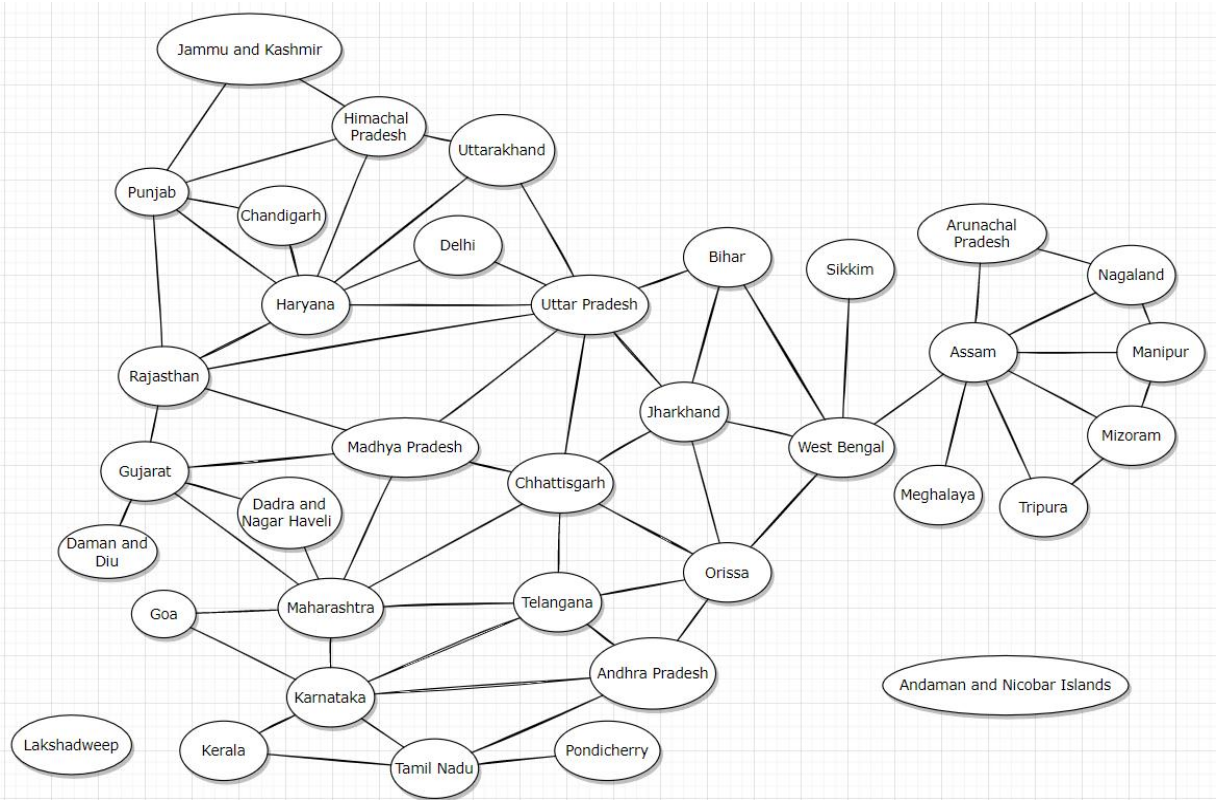
After this, I browsed all edges in this graph and I will stop finding R1 at here. So I found my initial circuit R1 which is UVbaU and it's in the last of TestCircuit.And then, I will assign all element of the TestCircuit to outCircuit.

Finally, I have the result which is outCircuit that is "U,T,Z,a,g,h,b,c,W,V, P,Q,W,X,d,c,i,h,m,l,g,f,Z,Y,S,T,N,M,S,B,Y,e,f,k,l,D,m,n,i,j,d,C,X,R,Q,L,

K,J,I,N,O,J,A,I,E,M,B,e,G,k,D,n,H,j,C,R,F,L,A,E,B,G,D,H,C,F,A,K,P,O,U,V,b,a,U".

CHAPTER 8 – MAP COLORING

8.1 Problem a



Picture 8.1 Graph of this map

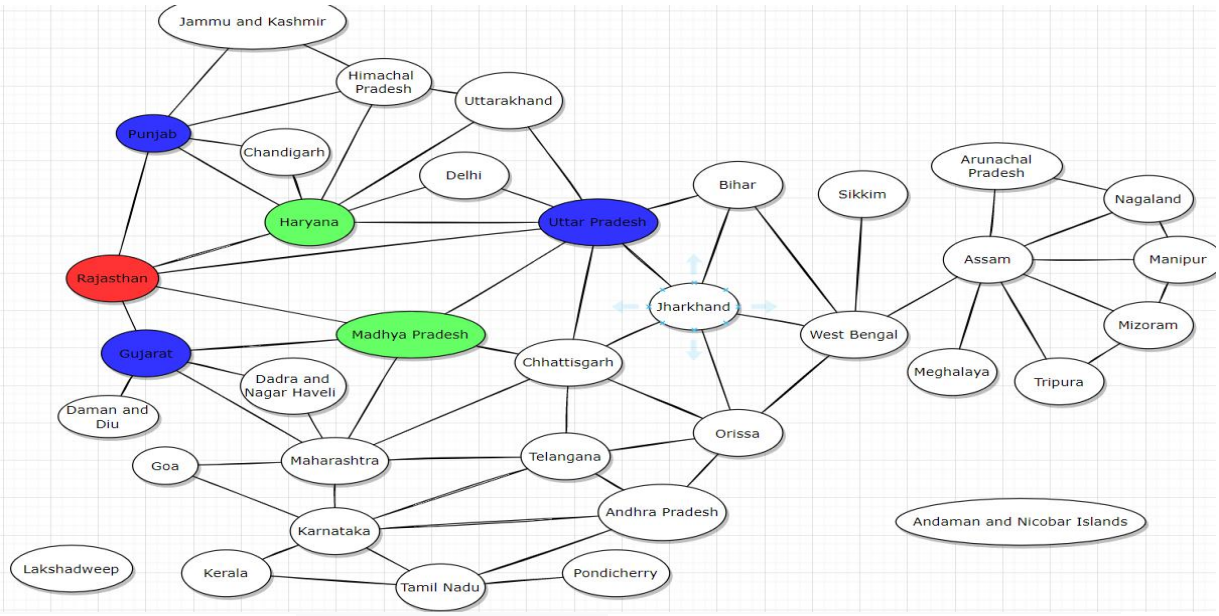
8.2 Problem b

My student id is 520H0418 so abcd = 0418, abcd % 4 so 0418%4 = 2 so I will start from Rajasthan.

A minimum number of colors is the algorithm which is coloring the graph but the edges is neighbour together which is not a same color.

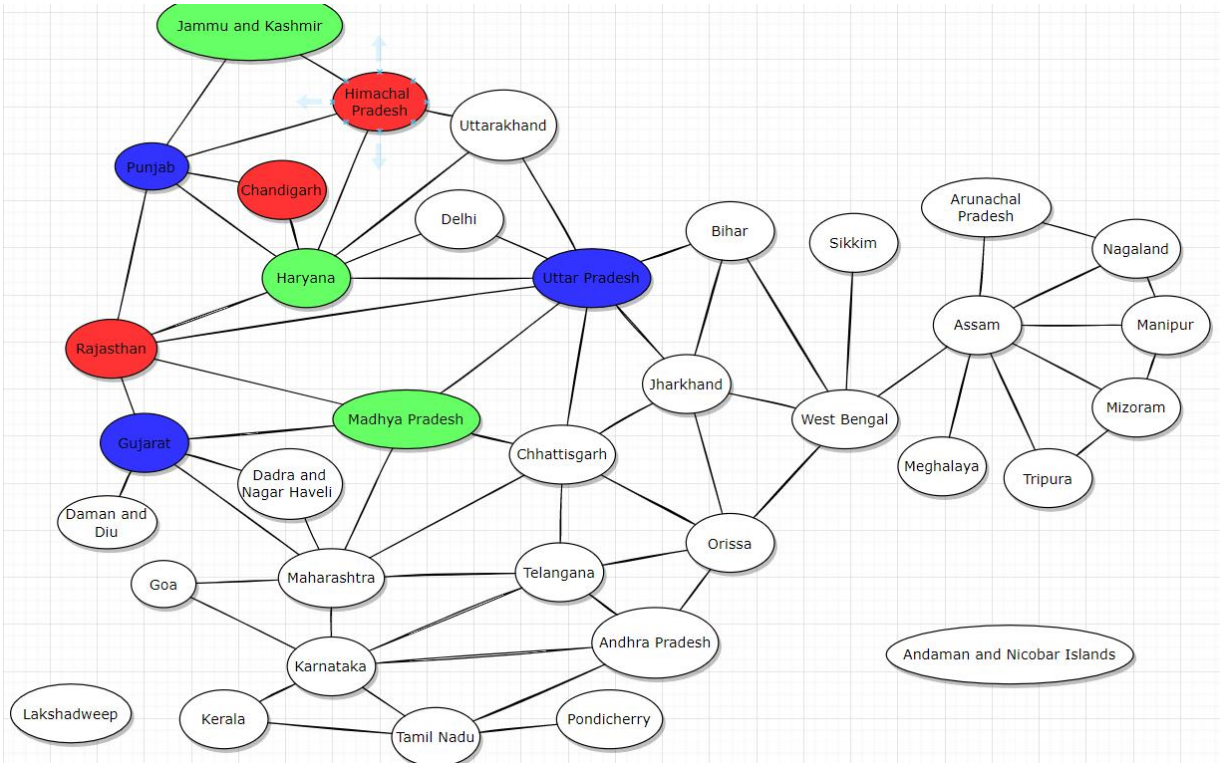
I will color this graph following step by step below:

So I will start from “Rajasthan” and I color it that is red color. The neighbour of “Rajasthan” vertices include “Punjab”, “Haryana”, “Uttar Pradesh”, “Madhya Pradesh”, “Gujarat” vertex so I will color the blue color to “Punjab” vertex. I will color the green color to “Haryana” vertex because it has the neighbour with “Rajasthan” and “Punjab” vertex. I will color the blue color to “Uttar Pradesh” vertex because it has the neighbour with “Rajasthan” and “Haryana” vertex. I will color the green color to “Madhya Pradesh” vertex because it has the neighbour with “Rajasthan” and “Uttar Pradesh” vertex. I will color the blue color to “Gujarat” vertex because it has the neighbour with “Rajasthan” and “Madhya Pradesh” vertex.



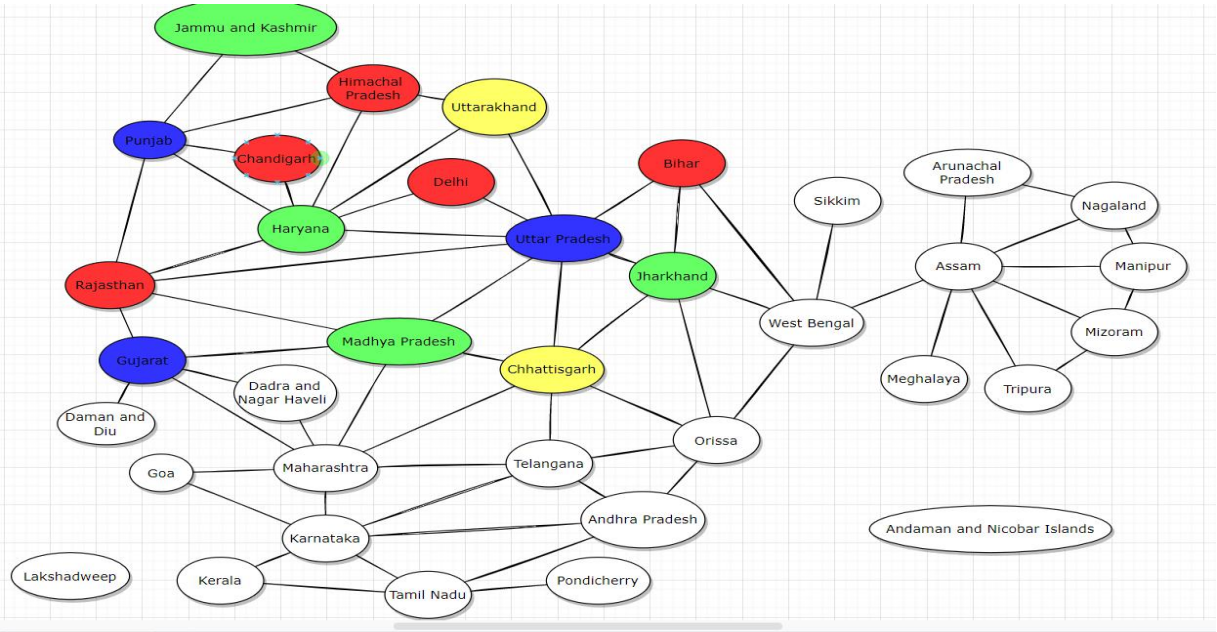
Picture 8.2.1 Coloring to “Rajasthan” vertex

Continue with “Punjab” vertex,which has the neighbour including “Jammu and Kashmir”, “Himachal Pradesh”, “Chandigarh”, “Haryana” and “Rajasthan”. I will color the green color to “Jammu and Kashmir” vertex. I will color the red color to “Himachal Pradesh” vertex because it has the neighbour with “Punjab”, “Jammu and Kashmir” and “Haryana” vertex. I will color the red color to “Chandigarh” vertex because it has the neighbour with “Punjab” and “Haryana” vertex. The Remaining neighbour vertices are colored.



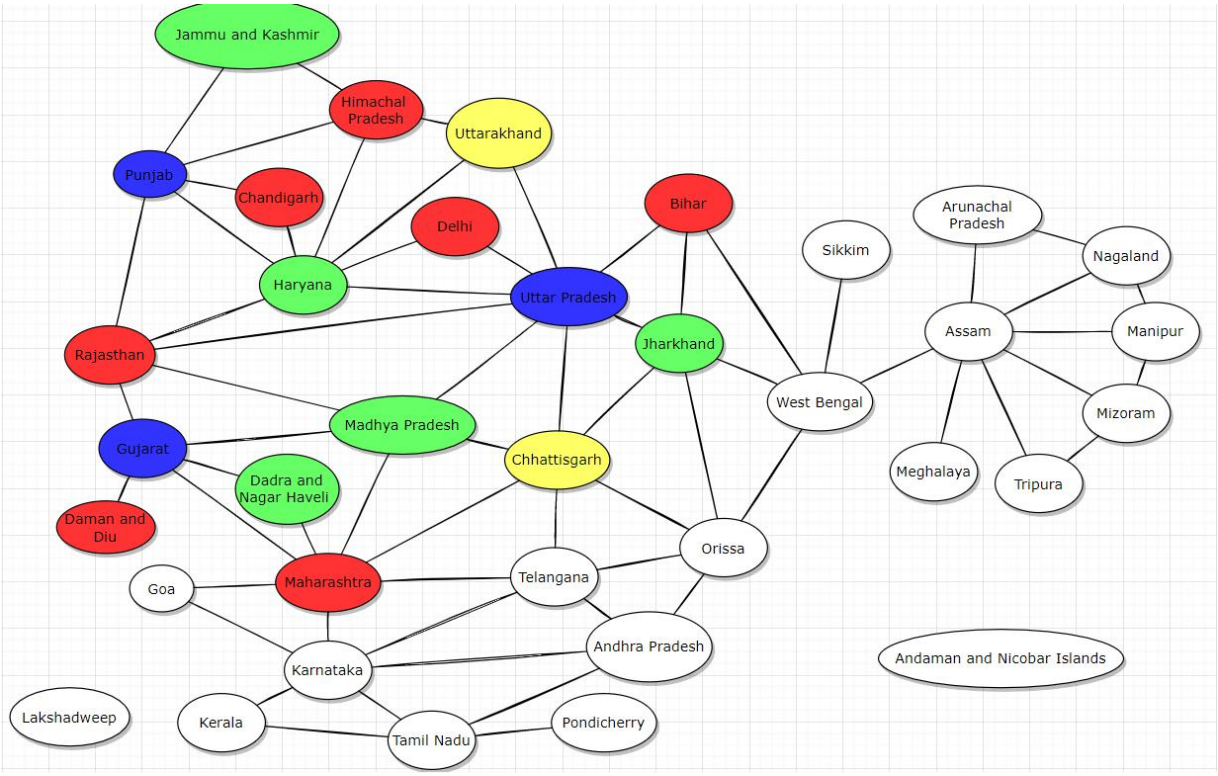
Picture 8.2.2 Coloring to “Punjab” vertex

Continue with “Uttar Pradesh” vertex,which has the neighbour including “Uttarakhand”, “Delhi”, “Madhya Pradesh”, “Chhattisgarh”, “Jharkhand”, “Bihar”, “Haryana” and “Rajasthan”. I will color the yellow color to “Uttarakhand” vertex because it has the neighbour including “Himachal Pradesh”, “Haryana” and “Uttar Pradesh” vertex. I will color the red color to “Delhi” vertex because it has the neighbour including “Haryana” and “Uttar Pradesh”. I will color the red color to “Bihar” vertex because it has the neighbour including “Uttarakhand”. I will color the green color to “Jharkhand” vertex because it has the neighbour including “Uttarakhand” and “Bihar” vertex. I will color the green color to “Madhya Pradesh” vertex because it has the neighbour including “Uttar Pradesh”, “Gujarat”, “Chhattisgarh”, “Maharashtra”, and “Rajasthan” vertex. I will color the yellow color to “Chhattisgarh” vertex because it has the neighbour including “Uttar Pradesh”, “Madhya Pradesh”, “Chhattisgarh”, “Maharashtra”, “Orissa”, and “Telangana” vertex. The remaining neighbour vertices of “Uttar Pradesh” vertex are colored.



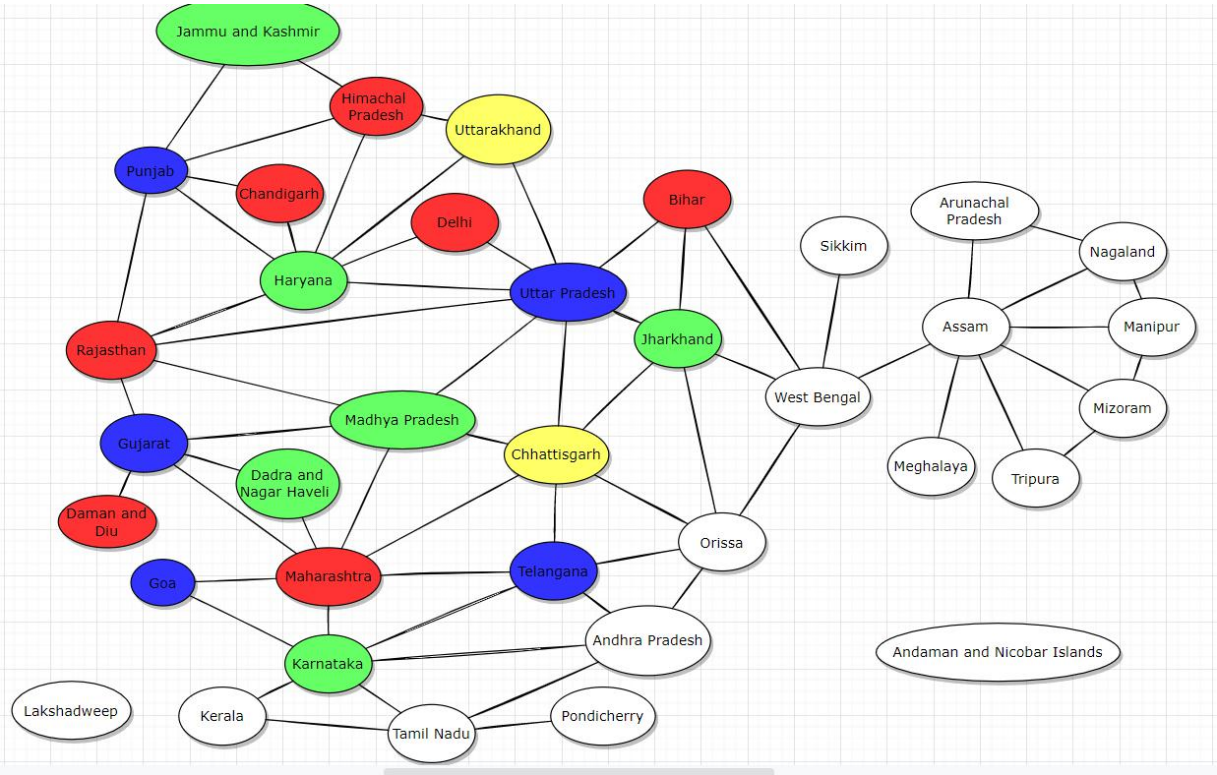
Picture 8.2.3 Coloring to “Uttar Pradesh” vertex

Continue with “Gujarat” vertex,which has the neighbour including “Daman and Diu”, “Dadra and Nagar Haveli”, “Madhya Pradesh”, “Maharashtra”, and “Rajasthan” vertex. I will color the yellow color to “Daman and Diu” vertex. I will color the red color to “Maharashtra” vertex because it has the neighbour including “Dadra and Nagar Haveli”, “Madhya Pradesh”, “Chhattisgarh”, “Telangana”, “Karnataka”, and “Goa” vertex. I will color the green color to “Dadra and Nagar Haveli” vertex because it has the neighbour including “Maharashtra” vertex. The remaining neighbour vertices of “Gujarat” vertex are colored.



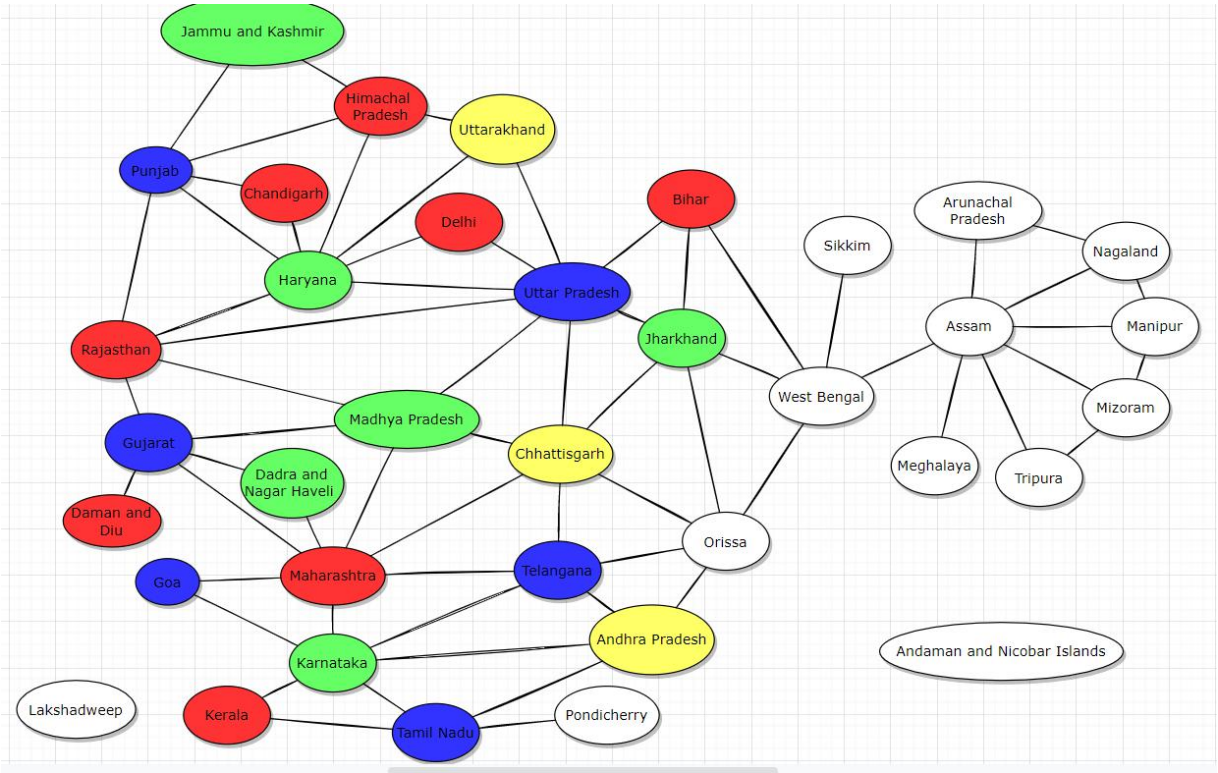
Picture 8.2.4 Coloring to “Gujarat” vertex

Continue with “Maharashtra” vertex,which has the neighbour including “Gujarat”, “Dadra and Nagar Haveli”, “Madhya Pradesh”, “Chhattisgarh”, “Karnataka”, “Goa” and “Telangana” vertex. I will color the blue color to “Telangana” vertex because it has the neighbour including “Maharashtra”, “Orissa”, “Chhattisgarh”, “Andhra Pradesh” and “Karnataka” vertex. I will color the green color to “Karnataka” vertex because it has the neighbour including “Maharashtra”, “Telangana”, “Goa”, “Andhra Pradesh”, “Tamil Nadu” and “Kerala” vertex. I will color the blue color to “Goa” vertex because it has the neighbour including “Maharashtra” and “Karnataka” vertex. The remaining neighbour vertices of “Maharashtra” vertex are colored.



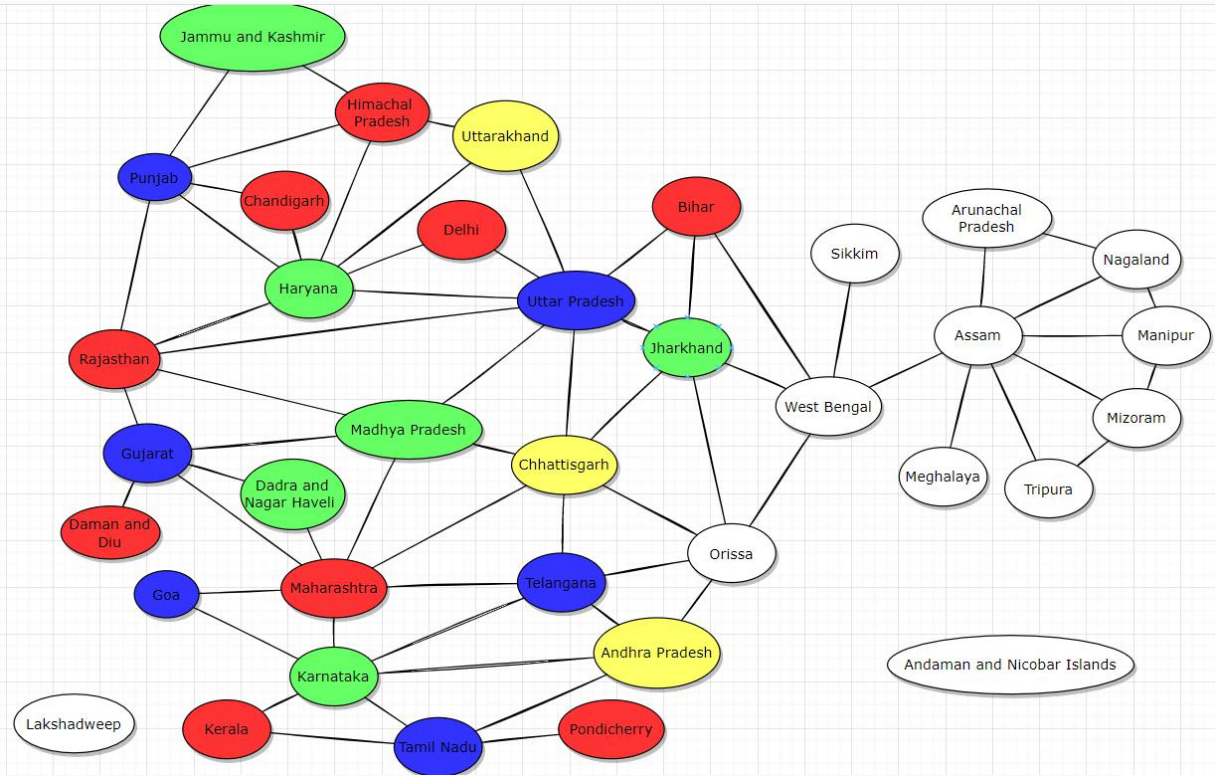
Picture 8.2.5 Coloring to “Maharashtra” vertex

Continue with “Karnataka” vertex,which has the neighbour including “Maharashtra”, “Telangana”, “Goa”, “Andhra Pradesh”, “Tamil Nadu” and “Kerala” vertex. I will color the yellow color to “Andhra Pradesh” vertex because it has the neighbour including “Telangana”, “Orissa”, “Tamil Nadu” and “Karnataka” vertex. I will color the blue color to “Tamil Nadu” vertex because it has the neighbour including “Andhra Pradesh”, “Kerala”, “Pondicherry” and “Karnataka” vertex. I will color the red color to “Kerala” vertex because it has the neighbour including “Tamil Nadu” and “Karnataka” vertex. The remaining neighbour vertices of “Karnataka” vertex are colored.



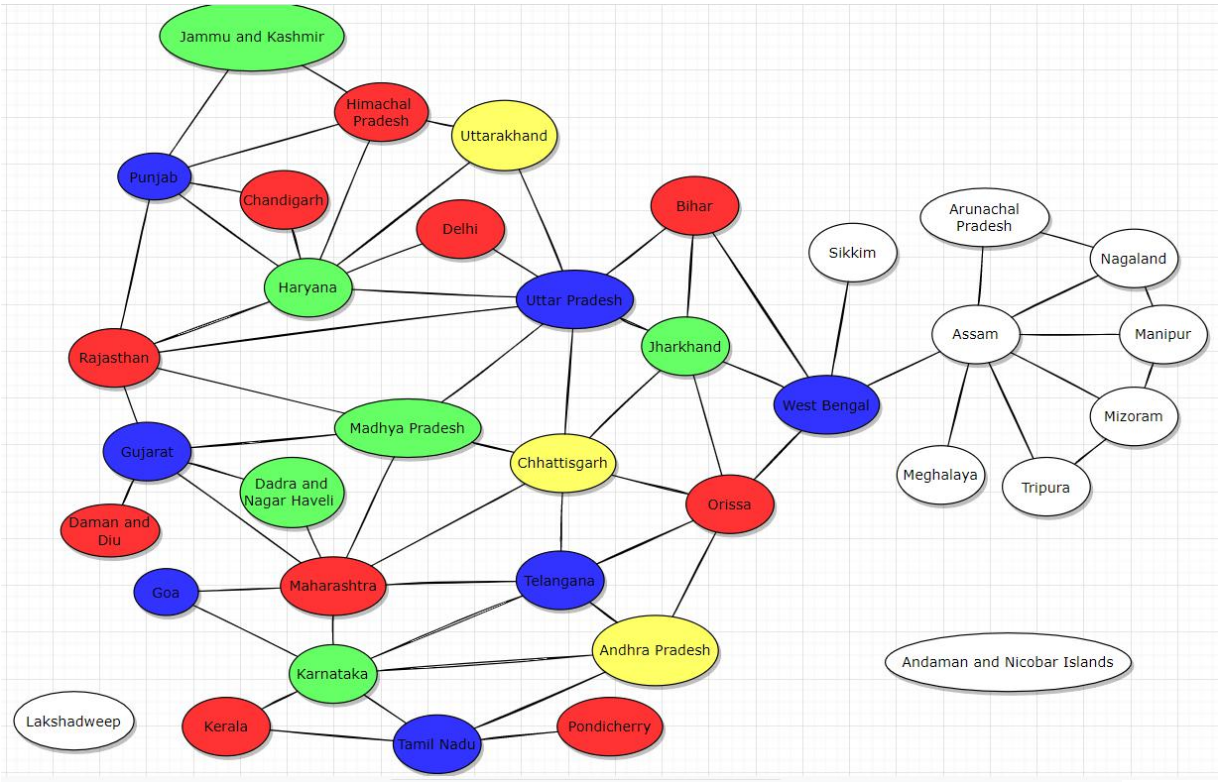
Picture 8.2.6 Coloring to “Karnataka” vertex

Continue with “Tamil Nadu” vertex,which has the neighbour including “Karnataka”, “Telangana”, “Andhra Pradesh” and “Kerala” vertex. I will color the red color to “Pondicherry” vertex. The remaining neighbour vertices of “Tamil Nadu” vertex are colored.



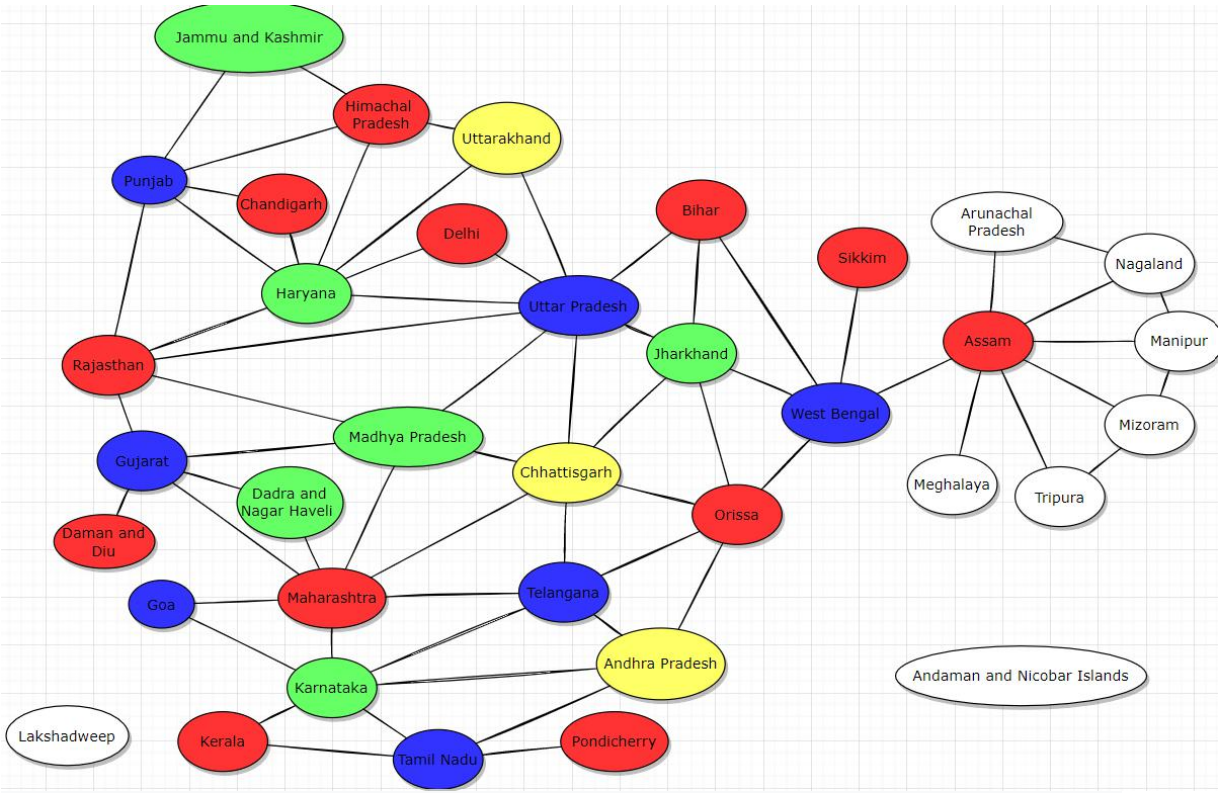
Picture 8.2.7 Coloring to “Tamil Nadu” vertex

Continue with “Jharkhand” vertex,which has the neighbour including “Uttar Pradesh”, “Bihar”, “West Bengal”, “Chhattisgarh” and “Orissa” vertex. I will color the red color to “Orissa” vertex because it has the neighbour including “Andhra Pradesh”, “Chhattisgarh”, “Jharkhand”, “West Bengal” and “Telangana” vertex. I will color the blue color to “West Bengal” vertex because it has the neighbour including “Bihar”, “Orissa”, “Jharkhand”, “Sikkim” and “Assam” vertex. The remaining neighbour vertices of “Jharkhand” vertex are colored.



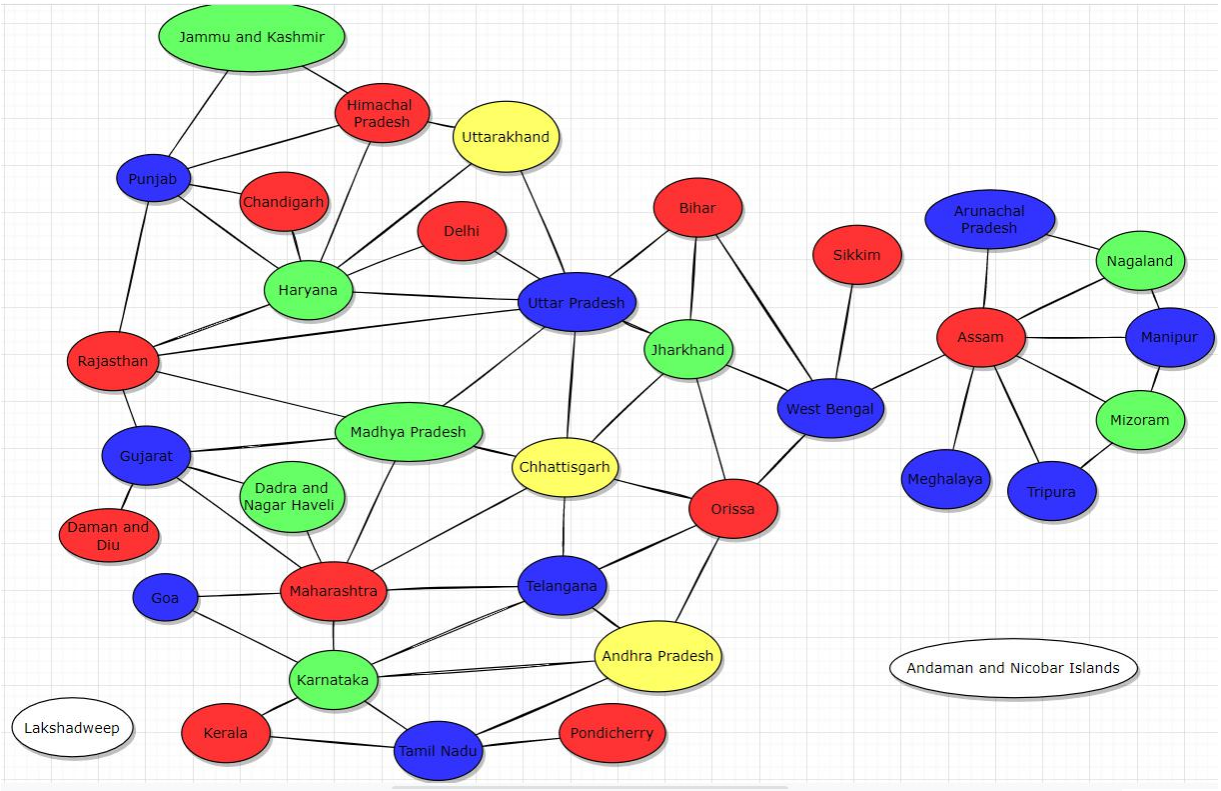
Picture 8.2.8 Coloring to “Jharkhand” vertex

Continue with “West Bengal” vertex,which has the neighbour including “Jharkhand”, “Bihar”, “Sikkim”, “Assam” and “Orissa” vertex. I will color the red color to “Sikkim” vertex. I will color the red color to “Assam” vertex because it has the neighbour including “Arunachal Pradesh”, “Nagaland”, “Manipur”, “West Bengal”, “Meghalaya”, “Tripura” and “Mizoram” vertex. The remaining neighbour vertices of “West Bengal” vertex are colored.



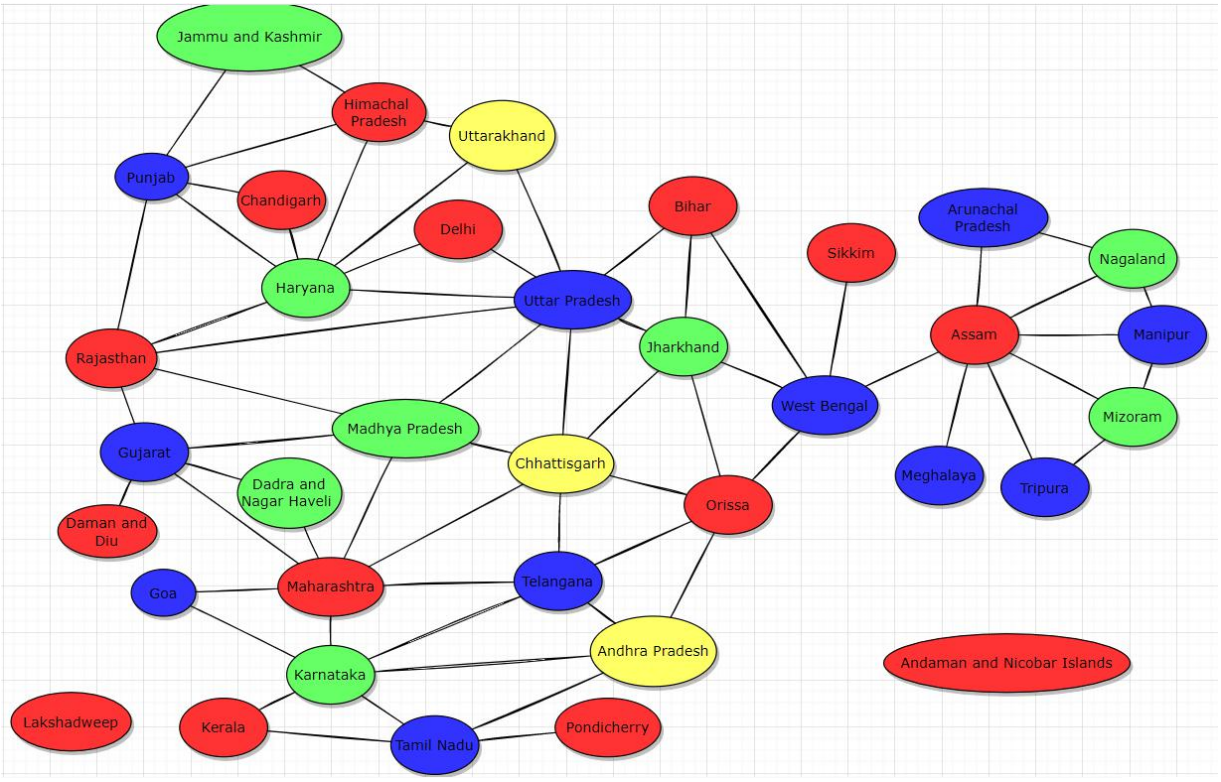
Picture 8.2.9 Coloring to “West Bengal” vertex

Continue with “Assam” vertex,which has the neighbour including “Arunachal Pradesh”, “Nagaland”, “Manipur”, “West Bengal”, “Meghalaya”, “Tripura” and “Mizoram” vertex. I will color the blue color to “Arunachal Pradesh” vertex. I will color the green color to “Nagaland” vertex because it has the neighbour including “Arunachal Pradesh”, “Assam” and “Manipur” vertex. I will color the blue color to “Manipur” vertex because it has the neighbour including “Nagaland”, “Assam” and “Mizoram” vertex. I will color the green color to “Mizoram” vertex because it has the neighbour including “Tripura”, “Assam” and “Manipur” vertex. I will color the blue color to “Tripura” vertex because it has the neighbour including “Assam” and “Mizoram” vertex. I will color the blue color to “Meghalaya” vertex. The remaining neighbour vertices of “Assam” vertex are colored.



Picture 8.2.10 Coloring to “Assam” vertex

Finally, I have the two remaining vertices including “Lakshadweep”, “Andaman and Nicobar Islands” vertex which are zero degree vertex so I will color it the red color.



Picture 8.2.11 Coloring to the two remaining vertices