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

BÁO CÁO CUỐI KÌ MÔN CẤU TRÚC RỜI RẠC

Người hướng dẫn: **TS NGUYỄN THỊ HUỲNH TRÂM**
Người thực hiện: **NGUYỄN TÔN ĐIỀN - 52000643**
Lớp : **20050201**
Khoá : **24**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

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

BÁO CÁO CUỐI KÌ MÔN CẤU TRÚC RỜI RẠC

Người hướng dẫn: **TS NGUYỄN THỊ HUỲNH TRÂM**
Người thực hiện: **NGUYỄN TÔN ĐIỀN - 52000643**
Lớp : **20050201**
Khoá : **24**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

TÓM TẮT

Báo cáo này giúp em hệ thống lại những kiến thức như: thuật toán Euclid mở rộng, giải bài toán quy nạp, các phép toán trên tập hợp, số nghịch đảo trong cấu trúc modulo, cấu trúc dữ liệu Union Find để kiểm tra chu trình trong thuật toán Kruskal, giải bài toán tô màu bản đồ với số màu ít nhất.

MỤC LỤC

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

CHƯƠNG 1 – CÂU HỎI SỐ 1

MSSV của em là 52000643, do đó m = 643

1.1 Tính gcd(2021, 1643) và lcm (2021,1643) bằng thuật toán Euclid

gcd(2021,1643)

2021 = 1643 x 1 + 378 ← gcd(1643,378)

1643 = 378 x 4 + 131 ← gcd(378,131)

378 = 131 x 2 + 116 ← gcd(131,116)

131 = 116 x 1 + 15 ← gcd(116,15)

116 = 15 x 7 + 11 ← gcd(15,11)

15 = 11 x 1 + 4 ← gcd(11,4)

11 = 4 x 2 + 3 ← gcd(4,3)

4 = 3 x 1 + 1 ← gcd(3,1)

3 = 1 x 3 + 0 ← gcd(1,0)

Do đó: gcd(2021,1643) = 1

Từ đó: lcm(2021,1643) = 2021 x 1643 / gcd(2021,1643)

= 2021 x 1643 / 1

= 3320503

lcm(2021,1643) = 3320503

1.2 Tìm 5 cặp nghiệm (x, y) cho phương trình sau:

2021x + 1643y = 1

Ta có:

1 = 4 – 3 x 1 = 4 x 3 x(-1) = 4 x (11 - 4 x 2) x (-1) = 11 x (-1) + 4 x 3

= 11 x (-1) + (15 – 11 x 1) x 3 = 15 x 3 + 11 x (-4)

= 15 x 3 + (116 – 15 x 7) x (-4) = 116 x (-4) + 15 * 31

= 116 x (-4) + (131 – 116 x 1) x 31 = 131 x 31 + 116 x (-35)

= 131 x 31 + (378 -131 x 2) x (-35) = 378 x (-35) + 131 * 101

= 378 x (-35) + (1643 – 378 x 4) x 101 = 1643 x 101 + 378 x (-439)

= 1643 x 101 + (2021 – 1643 x 1) x (-439)

= 2021 x (-439) + 1643 x 540

Theo tính chất B’ezout:

$(x_1, y_1) = (-439, 540)$

$ax + by = d$

$(x, y) = \left(x + \frac{kb}{d}, y - \frac{ka}{d}\right)$

Chọn k = 1

$$(x_2, y_2) = \left(-439 + \frac{1 \times 1643}{1}, 540 - \frac{1 \times 2021}{1}\right) = (1204, -1481)$$

Chọn k = 2

$$(x_3, y_3) = \left(-439 + \frac{2 \times 1643}{1}, 540 - \frac{2 \times 2021}{1}\right) = (2847, -3502)$$

Chọn k = 3

$$(x_4, y_4) = \left(-439 + \frac{3 \times 1643}{1}, 540 - \frac{3 \times 2021}{1}\right) = (4490, -5523)$$

Chọn k = 4

$$(x_5, y_5) = \left(-439 + \frac{4 \times 1643}{1}, 540 - \frac{4 \times 2021}{1}\right) = (6133, -7544)$$

CHƯƠNG 2 – CÂU HỎI SỐ 2

2. Giải bài toán quy nạp sau

$a_n = 8.a_{n-1} - 15.a_{n-2}$ với $a_0 = 5$ và $a_1 = 43$

Chuỗi trên thỏa mãn giả thuyết của hệ thức truy hồi tuyến tính thuần nhất bậc 2 với hệ số hằng là A = 8 và B = - 15. Khi đó, phương trình đặc trưng của chuỗi trên là $t^2 - At - B = 0$ hay $t^2 - 8t + 15 = 0$ (1)

Nghiệm của phương trình đặc trưng là $t = \frac{8 \pm \sqrt{(-8)^2 - 4 \times (1 \times 15)}}{2} = \begin{cases} r = 5 \\ s = 3 \end{cases}$(2)

Khi đó 2 nghiệm này thỏa mãn phương trình nghiệm tổng quát:

$a_n = C \times 5^n + D \times 3^n$ (3)

Với $a_0 = 5$ và $a_1 = 43$

$a_0 = 5 = C \times 5^0 + D \times 3^0 = C \times 1 + D \times 1 = C + D$

$a_1 = 43 = C \times 5^1 + D \times 3^1 = C \times 5 + D \times 3 = 5C + 3D$

$\begin{cases} C + D = 5 \\ 5C + 3D = 43 \end{cases} \Rightarrow \begin{cases} C = 14 \\ D = -9 \end{cases}$

Thay C, D vào (3)

$a_n = 14 \times 5^n - 9 \times 3^n$

CHƯƠNG 3 – CÂU HỎI SỐ 3

3.1 Tạo tập hợp Γ từ những chữ cái trong tên của bản thân:

Họ và tên: Nguyễn Tôn Điền, do đó $\Gamma = \{D, E, G, I, N, O, T, U, Y\}$

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

3.2 Các phép toán trên tập hợp:

$\Gamma \cup \Delta = \{A, C, D, E, G, H, I, N, O, T, U, Y\}$

$\Gamma \cap \Delta = \{D, G, N, O, T, U\}$

$\Gamma \setminus \Delta = \{E, I, Y\}$

$\Gamma \ominus \Delta = \{A, C, E, H, I, Y\}$

CHƯƠNG 4 – CÂU HỎI SỐ 4

MSSV của em là 52000643, m = 43

Gọi R là mối quan hệ nhị phân được định nghĩa bởi 2 số tự nhiên dưới đây:

$\forall a, b \in N\ (aRb \leftrightarrow 43|(a.b))$

4.1 Xét tính phản xạ của R (reflexive):

Để chứng minh R phản xạ (reflexive), ta cần phải chỉ ra rằng:

$\forall a \in N, aRa$

Với định nghĩa của R, điều trên có nghĩa là:

$\forall a \in N, 43|(a.a)$

Điều trên là sai với a = 7, a.a = 49 (49 không chia hết 43)

Do đó, R không phản xạ

4.2 Xét tính đối xứng của R (symetric):

Để chứng minh R đối xứng (symetric), ta cần chỉ ra:

$\forall a, b \in N$, nếu aRb thì bRa

Với định nghĩa của R, điều trên có nghĩa là:

$\forall a, b \in N$, nếu $43|(a.b)$ thì $43|(b.a)$

Điều trên là hoàn toàn đúng bởi a.b = b.a bởi tính giao hoán của phép nhân

Do đó, R đối xứng

4.3 Xét tính phản đối xứng của R (anti-symetric):

Để chứng minh R phản đối xứng (anti-symetric), ta phải chỉ ra:

$\forall a \in N, \forall b \in N$, nếu (aRb) và (bRa) thì a = b

Với định nghĩa của R, điều trên có nghĩa là:

$\forall a \in N, \forall b \in N$, nếu $43|(a.b)$ và $43|(b.a)$ thì a = b

Điều trên là sai với a = 43, b = 2, $43|(43.2)$ và $43|(2.43)$ nhưng $43 \neq 2$

Do đó, R không phản đối xứng

4.4 Xét tính bắc cầu của R (transitive):

Để chứng minh R bắc cầu (transitive), ta cần chỉ ra:

$\forall a, b, c \in N$, nếu aRb và bRc thì aRc

Với định nghĩa của R, điều trên có nghĩa là:

$\forall a, b, c \in N$, nếu $43|(a.b)$ và $43|(b.c)$ thì $43|(a.c)$ (*)

Với định nghĩa của phép chia hết ta có, $a.b = 43.r$ với r là một số nguyên, và $b.c = 43.s$ với s là một số nguyên

Ta thấy rằng, $(a.b) . (b.c) = a.c.b^2 = 43r . 43s$

Hay $a.c = 43 [(43.r.s)/b^2]$

Ta có thể chỉ ra, với $r = 2$, $s = 3$, $b = 4$, $(43.r.s)/b^2 = (43.2.3)/4^2 = 16.125$ (không phải số nguyên)

Theo định nghĩa của phép chia hết thì khi $r = 2$, $s = 3$, $b = 4$, (a.c) không chia hết cho 43, đồng nghĩa (*) sai.

Do đó, R không bắc cầu

CHƯƠNG 5 – CÂU HỎI SỐ 5

5.1 Trình bày kiến thức về thuật toán Euclid mở rộng để tính các số nghịch đảo theo modulo:

Ta cần tìm x với: $x \equiv a^{-1} \pmod{n}$

Với bất kì số a, n thuộc tập số nguyên và n > 1, nếu không thỏa điều kiện này kết luận ngay không tìm được x, nếu thỏa thực hiện những bước sau:

- Theo định lí 4.7.3: với mọi a, chỉ tồn tại $a^{-1} \pmod{n}$, khi và chỉ khi, a, n là số nguyên tố cùng nhau. Nói cách khác, khi và chỉ khi gcd(a,n) = 1, mới tồn tại x.

- Nếu gcd(a,n) = 1, xét:

  + Theo tính chất B'ezout, tồn tại các số nguyên s, t sao cho: as + nt = 1.

  + Theo đó: as = 1 – nt (chuyển vế đổi dấu)

  + Theo định lí 8.4.1 (Epp), as = 1 – nt tương đương $as \equiv 1 \pmod{n}$ (*)

  + Do đó, $s \equiv x \equiv a^{-1} \pmod{n}$

  + Bài toán qui về tìm s, ta dùng thuật toán Euclid mở rộng để tìm được cặp nghiệm (s,t) đầu tiên cho phương trình as + nt = 1

***Giải thích (*), ta có định lí 8.4.1 (Epp) phát biểu như sau*:**

Với a; b, và n là những số nguyên và giả sử n > 1, tất cả những biểu thức dưới đây đều tương đương với nhau:

1. *n | (a - b)*
2. *$a \equiv b \pmod{n}$*
3. *$a = b + kn$ ($k \in Z$)*
4. *a và b có cùng số dư (không âm) khi chia cho n*
5. *a mod n = b mod n*

Biểu thức 2 tương đương biểu thức 3, với a = as, b = 1, k = -t, n = n, ta có: as = 1 – nt tương đương $as \equiv 1 \pmod{n}$.

5.2 Áp dụng ý tưởng trên để giải bài toán sau:

Với MSSV là 52000643, nên m = 43

Bài toán tìm: 44^{-1} (mod 101)

Tổng quát hóa, a = 44, n = 101, kết quả của bài toán 44^{-1} (mod 101) là x cần tìm.

Vì a, n đều là số nguyên và n > 1, ta đến bước tìm gcd(a,n) bằng giải thuật euclid

gcd(a,n) = gcd(n,a)= gcd(101,44)

101 = 44x2 + 13 ← gcd(44,13)

44 = 13x3 + 5 ← gcd(13,5)

13 = 5 x 2 + 3 ← gcd(5,3)

5 = 3 x 1 + 2 ← gcd(3,2)

3 = 2 x 1 + 1 ← gcd(2,1)

2 = 1 x 2 + 0 ← gcd(1,0)

Do đó, gcd(a,n) = 1, ta đến bước sử dụng thuật toán Euclid mở rộng để tìm cặp nghiệm (s,t) đầu tiên cho phương trình as + nt = 1 hay 44s + 101t = 1

1 = 3 – 2 = 3 – (5 - 3) = 3 – 5 + 3 = 2x3 – 5 = 2x3 – (44 – 13x3)

= 2x3 – 44 + 13x3 = 2x3 – 44 + (101 – 44x2)x3

= 2x3 – 44 + 101x3 – 44x6 = 2x3 – 44x7 + 101x3

= 2x(13 – 5x2) – 44x7 + 101x3 = 2x13 – 5x4 – 44x7 + 101x3

= 2x(101 – 44x2) – (44 – 13x3)x4 – 44x7 + 101x3

= 101x5 – 44x15 + 13x12 = 101x5 – 44x15 + (101 – 44x2)x12

= 101x17 – 44x39

Do đó, s ≡ 44^{-1} (mod 101) ≡ - 39. Áp dụng B'ezout, để có kết quả dương, ta có: -39 + 101 = 62 ($s_2 = s_1 + 1.t$), vậy 44^{-1} (mod 101) ≡ 62

CHƯƠNG 6 – CÂU HỎI SỐ 6

Lý thuyết dưới đây được tham khảo từ [1]

6.1. Giới thiệu Union Find để kiểm tra chu trình trong giải thuật Kruskal:

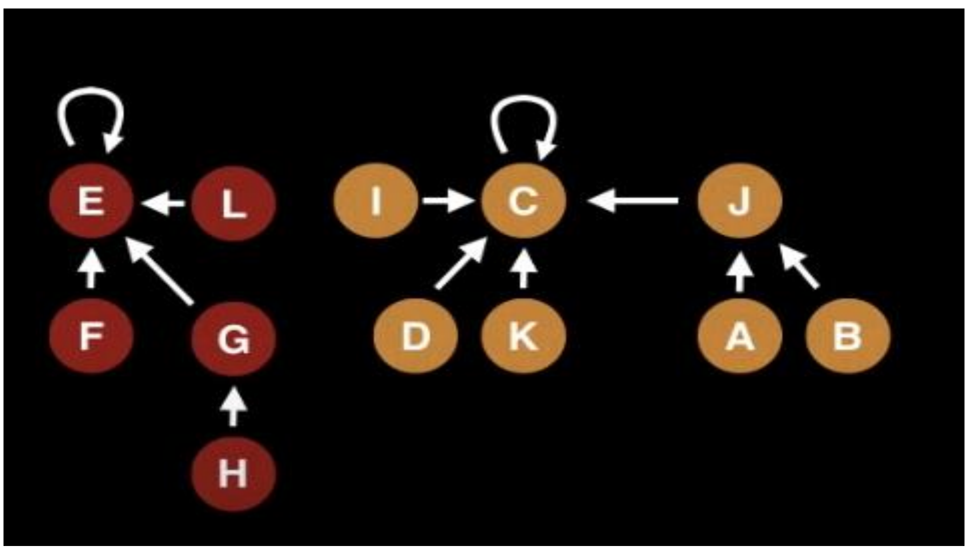
Giải thuật Kruskal là một phương pháp để tìm Minimum Spanning Tree (một graph con có tất cả các đỉnh của graph cha, không có chu trình và có tổng trọng số bé nhất), chính vì thế việc kiểm tra chu trình trong giải thuật là một việc hết sức quan trọng.

Để giải quyết vấn đề kiểm tra chu trình, Union Find là một phương pháp rất hiệu quả. Như cái tên của nó, giải thuật gồm 2 thao tác chính:

- Find(A): Nhằm tìm tập hợp chứa đỉnh A, trả về root của tập ấy

- Union(x, y): Gộp 2 tập hợp x, y lại bằng cách nối 2 phần tử root của mỗi tập hợp lại với nhau, từ đó tạo tập hợp mới lớn hơn.

Minh họa:



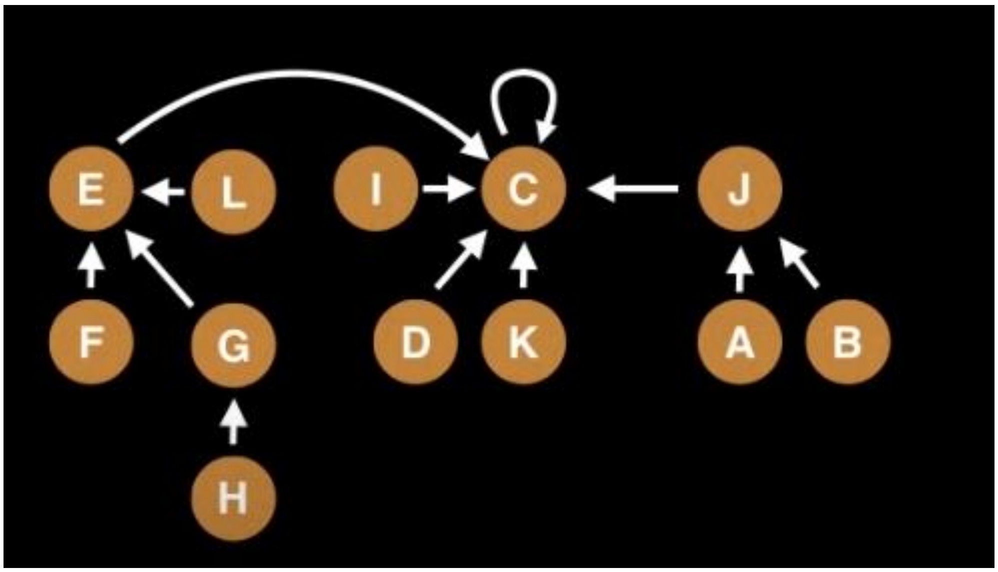
Hình 6.1 Minh họa về Find và Union

Find(H): Tìm tập hợp chứa đỉnh H, đó chính là tập hợp những đỉnh màu đỏ với root của tập hợp đó là đỉnh E. Nên find(H) = E.

Lưu ý thêm: quá trình tìm ra root E của find(H) bằng cách từ đỉnh H dò ngược lên đỉnh cha của nó là đỉnh G, từ đỉnh G dò lên cha của nó là đỉnh E mà đỉnh E lại trỏ về chính nó thế là ta kết luận root của tập chứa H là E.

Tương tự find(B) = C

Union(E,C): gộp 2 tập hợp các đỉnh màu đỏ và vàng lại bằng cách nối root E với root C. Ta được kết quả như sau:



Hình 6.2. Kết quả sau khi union(E,C)

6.2 Áp dụng Union Find vào thuật toán Kruskal

Ý tưởng, sau bước đầu tiên của thuật toán Kruskal ta có tập hợp EL là danh sách các cạnh được sắp xếp theo trọng số từ bé đến lớn. Ta duyệt tất cả các cạnh trong EL, nếu 2 đỉnh tạo nên cạnh đang xét, cùng thuộc 1 tập con, kết luận có chu trình.

Mã giả cho thuật toán Kruskal ở giai đoạn kiểm tra chu trình:

For each <cạnh tạo bởi 2 đỉnh u và v> in <EL>{

If(find(u) == find(v)){

//2 đỉnh u, v đều thuộc 1 tập hợp -> có chu trình

//bỏ qua

}else{

Union(x, y) *//với x, y là 2 root của 2 tập chứa mỗi đỉnh u và v*

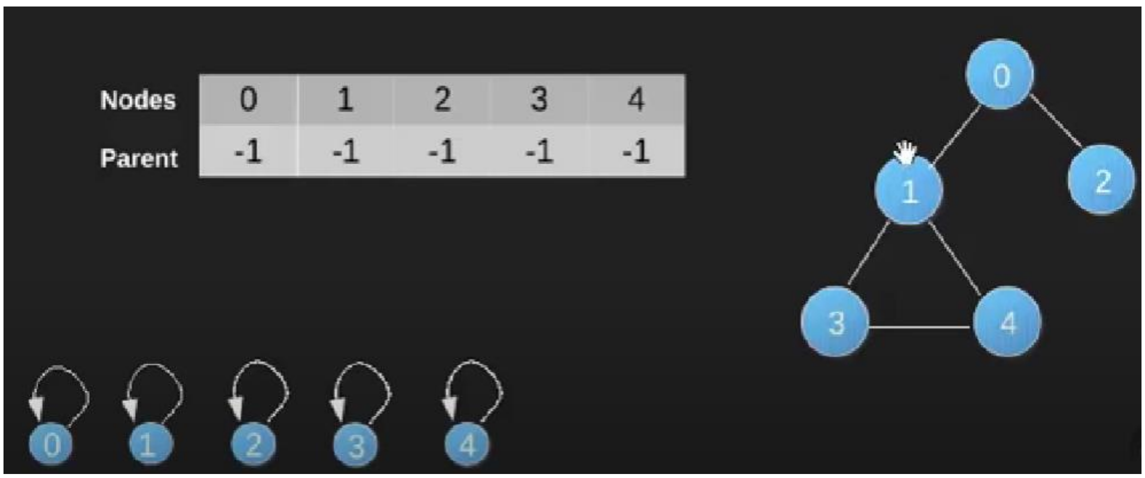
//đồng thời khi không tạo chu trình, thêm cạnh tạo bởi 2 đỉnh u và v vào MST mà ta đang xây dựng bằng thuật Kruskal

}

}

6.3. Minh họa trực quan qua ví dụ:

Không bàn đến thao tác thêm cạnh vào MST trong thuật toán Kruskal. Ta chỉ tập trung kiểm tra sự tồn tại chu trình của graph không trọng số dưới đây:

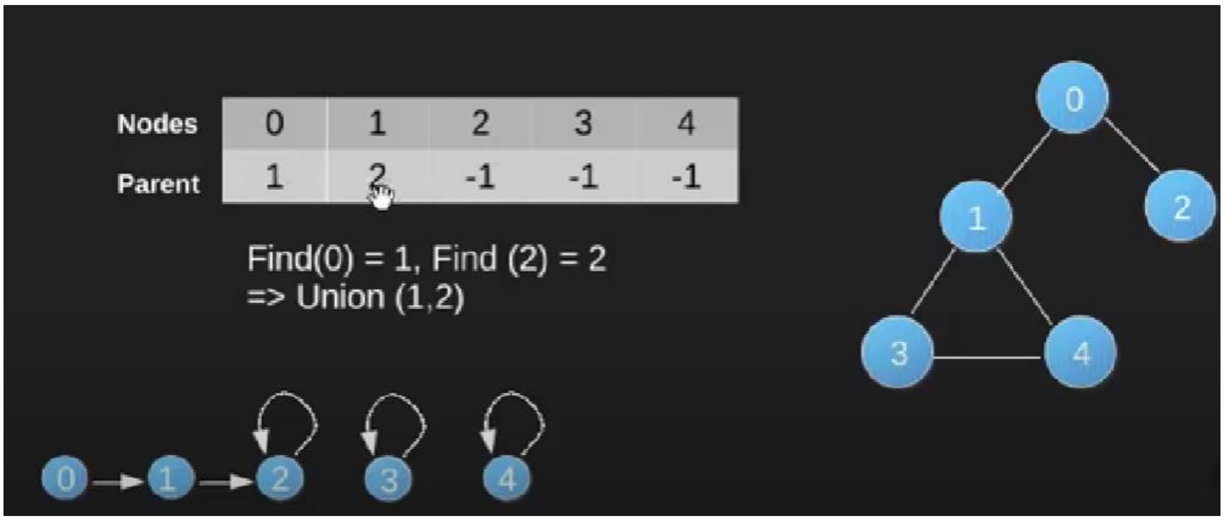


Hình 6.3. Các giá trị khởi tạo ban đầu của ví dụ

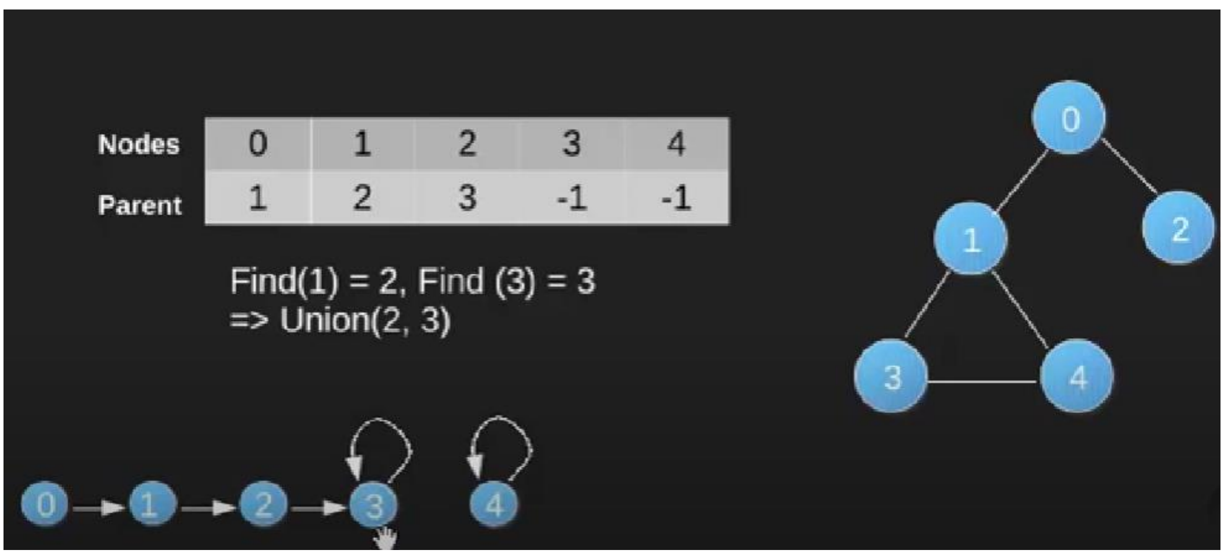
Hình 6.4. Sau lần chạy đầu tiên

Sau lần chạy đầu tiên, find(0) sẽ trả kết quả là 0 (giá trị đỉnh root của tập chứa đỉnh 0) và find(1) sẽ trả kết quả là 1. Vì find(0) ≠ find(1) ta union(0,1) được tập mới chứa 2 phần tử là 0 và 1 và có root là 1.

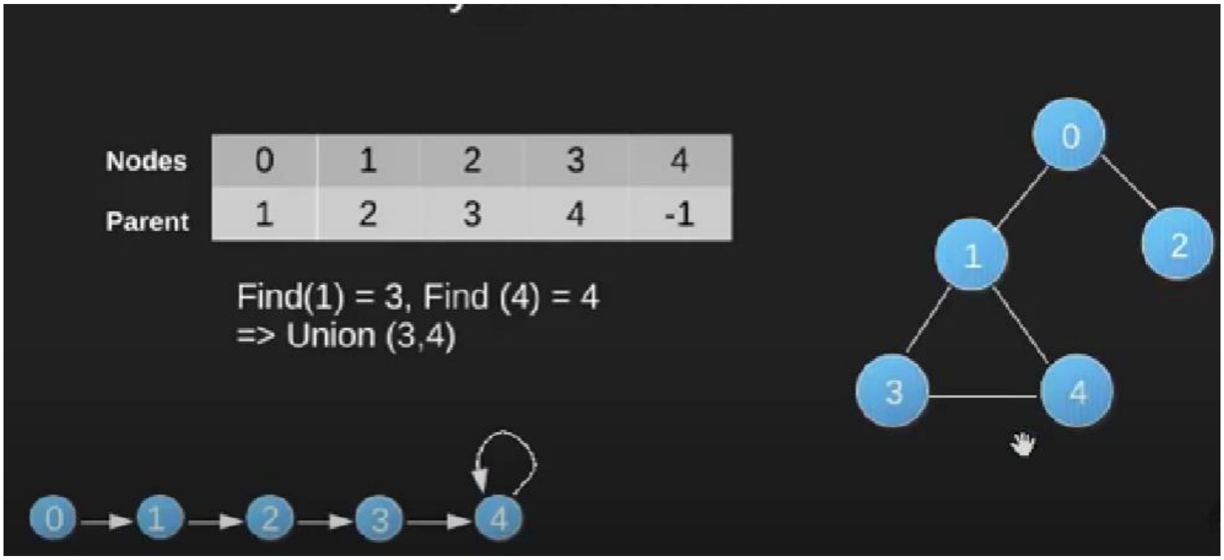
Tương tự với cách giải thích trên, ta có những kết quả sau:



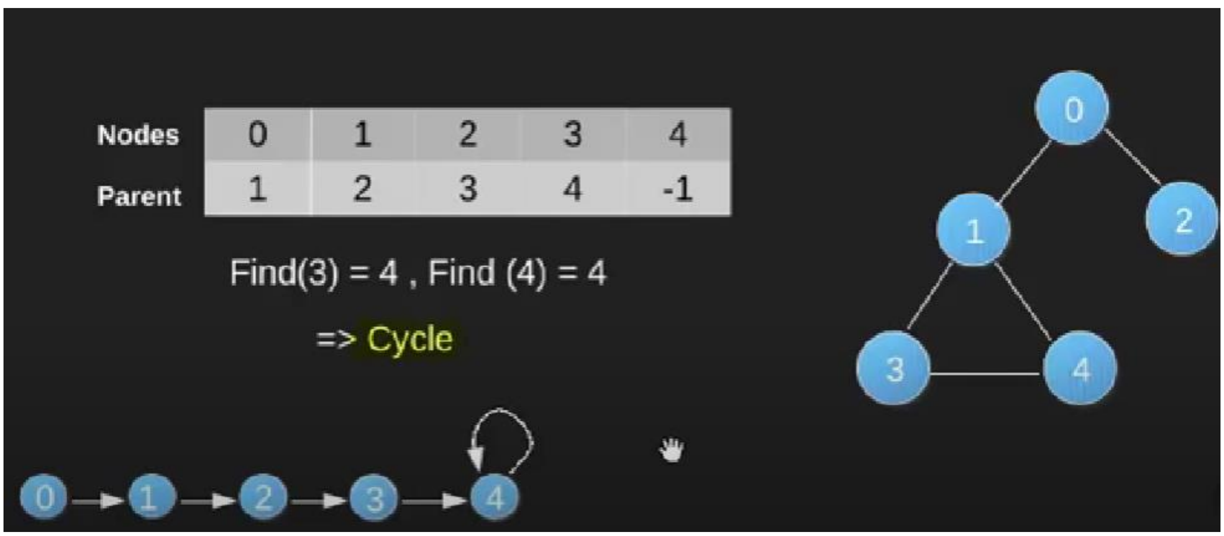
Hình 6.5. Kết quả sau lần chạy thứ 2



Hình 6.6. Kết quả sau lần chạy thứ 3



Hình 6.7. Kết quả sau lần chạy thứ 4

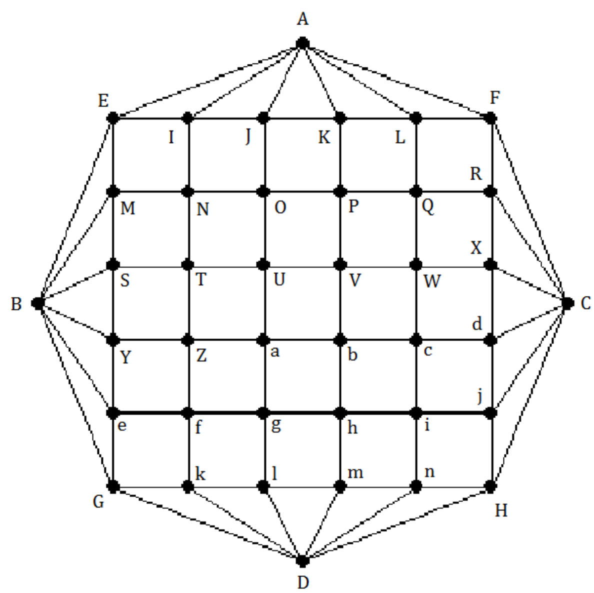


Hình 6.8. Kết quả cuối cùng

Vì đỉnh 3 và đỉnh 4 đều nằm chung trong tập hợp có root là 4 nên có thể kết luận graph trên có chu trình.

CHƯƠNG 7 – CÂU HỎI SỐ 7

7.1 Hình dưới có Eulerian circuit, Eulerian Path hay không?



Hình 7.1 Đề bài câu hỏi số 7

Ta nhận thấy tất cả các đỉnh trong hình 7.1 đều có bậc chẵn (4 và 6), thế nên đây không thể nào là Euler path vì để có Euler path phải có 2 đỉnh (đầu và cuối) bậc lẻ.

Đồ thị trên còn là đồ thị liên thông (connected graph)

Theo định lí 10.2.3, hình trên có Euler circuit

7.2 Trình bày về thuật toán Hierholzer để tìm Euclid circuit?

Lý thuyết và code dưới đây được tham khảo từ [2]

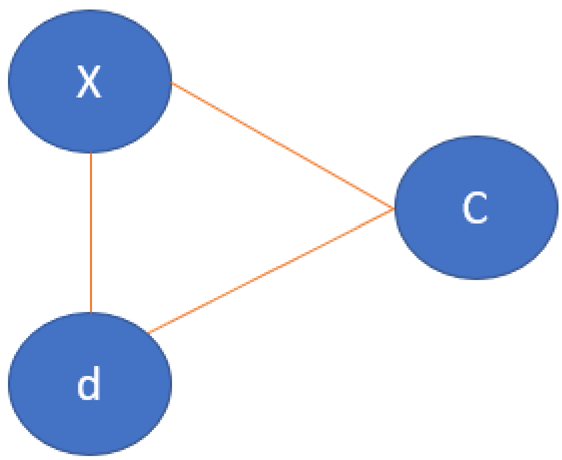
Các bước như sau:

1. Phải chắc hẳn đồ thị G có tồn tại Euclid circuit (không có đỉnh bậc lẻ)
2. Khởi tạo 2 stack. Stack đầu tên cpath, làm nhiệm vụ là stack nhớ tạm. Stack chính tên epath, lưu kết quả cuối cùng
3. Gọi v là đỉnh bắt đầu, vì tất cả đỉnh đều có bậc chẵn nên v là đỉnh bất kì.
4. Push v vào cpath (cpath.push(v)). Sau đó duyệt những đỉnh khác bằng DFS
5. Gọi u là đỉnh top của cpath stack (u = cpath.pop())

6. Nếu mọi cạnh từ u đều được duyệt, pop u từ cpath và push vào epath
7. Nếu u có các cạnh chưa được duyệt, chọn ngẫu nhiên 1 cạnh (u,x). Push x vào cpath và xóa cạnh (u,x) từ G.
8. Lặp lại bước 5 đến 7 cho đến khi cpath rỗng.

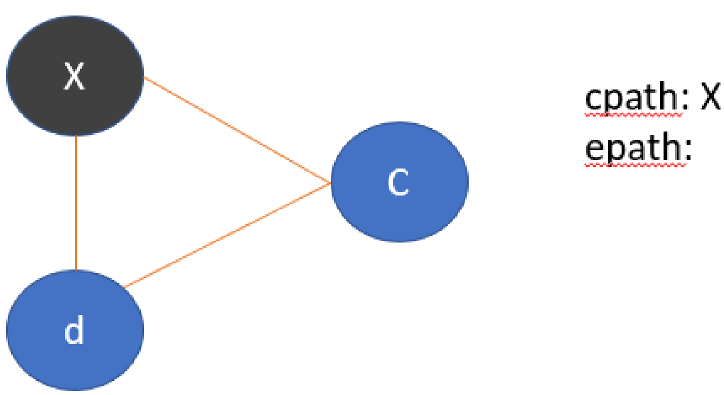
Euler circuit là kết quả của epath.

Minh họa ví dụ: Ta sẽ tìm Euler circuit graph với 3 đỉnh X, C, d:



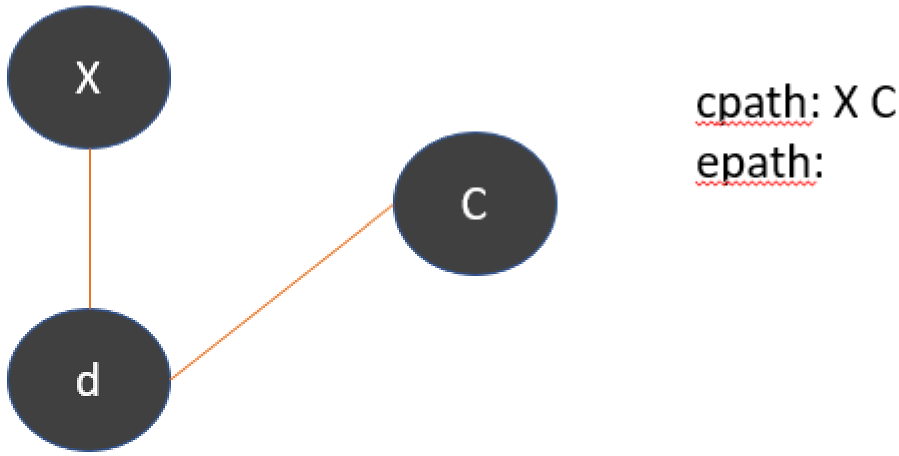
Hình 7.1. Ví dụ minh họa

- Khởi tạo: 2 stack và chọn đỉnh xuất phát là X, push X vào cpath



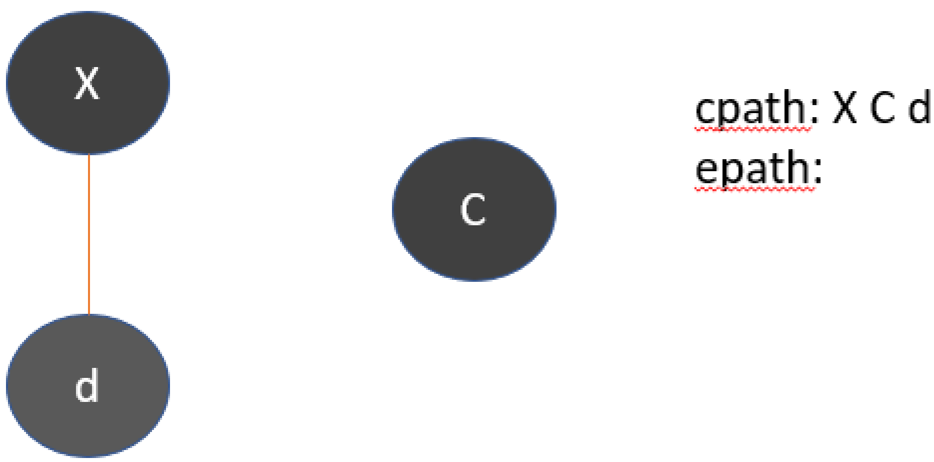
Hình 7.2. Khởi tạo

- Chạy lần 1, ta duyệt ngẫu nhiên cạnh (X,C) , push C vào cpath. Xóa (X,C)



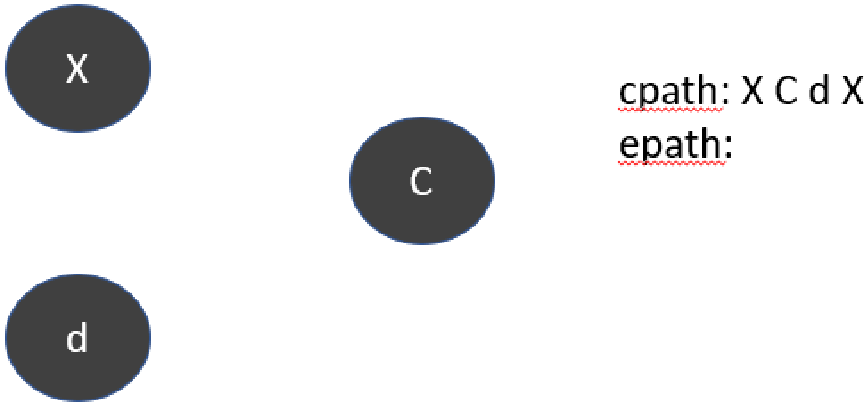
Hình 7.3 Chạy lần 1

- Chạy lần 2, top của cpath là C, ta chọn ngẫu nhiên cạnh bất kì từ C, ở đây chỉ còn cạnh (C,d), ta push d vào cpath, xóa (C,d)



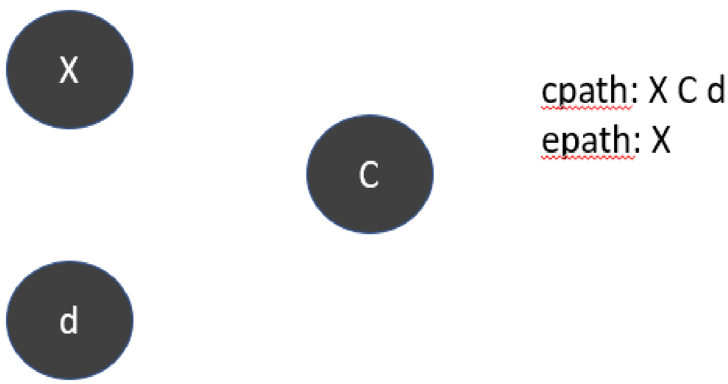
Hình 7.3 Chạy lần 2

- Chạy lần 3, top của cpath là d, ta duyệt cạnh duy nhất (d,X), push X và cpath, xóa (d,X)



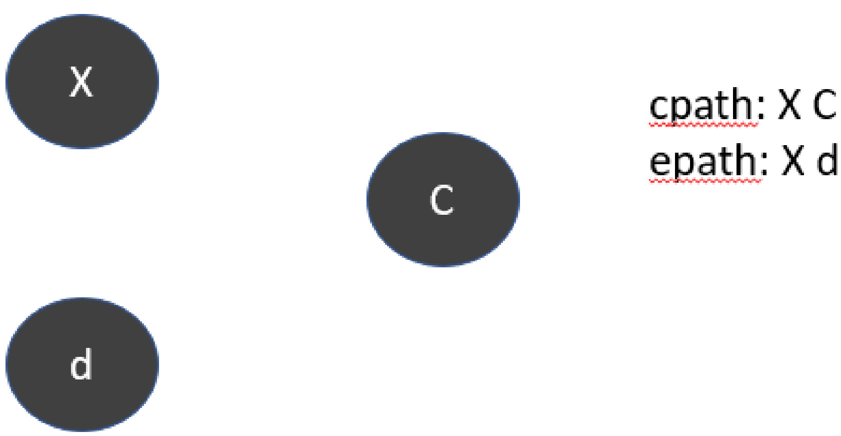
Hình 7.4 Chạy lần 3

- Chạy lần 4, top của cpath là X, X hết cạnh để duyệt, pop X từ cpath vào epath



Hình 7.5 Chạy lần 4

- Chạy lần 5, top của cpath là d, d hiết cạnh để duyệt, pop d từ cpath vào epath



Hình 7.6 Chạy lần 5

- Tương tự, ta check top của cpath nếu đỉnh đang xét không còn cạnh để duyệt thì push đỉnh đó cpath vào epath

Do đó kết quả cuối, epath: X d C X

7.3 Áp dụng thuật toán Hierholzer?

MSSV = 52000643, R1 = XCdX

Khởi tạo 2 stack và chọn đỉnh xuất phát

```
stack<int> cpath;    // current path
stack<int> epath;    // euler path

cpath.push(v);         // euler path starts from v
```

Hình 7.7 Code Hierholzer 1

Vòng lặp từ bước 5 đến bước 7 (đã được đề cập đầu chương)

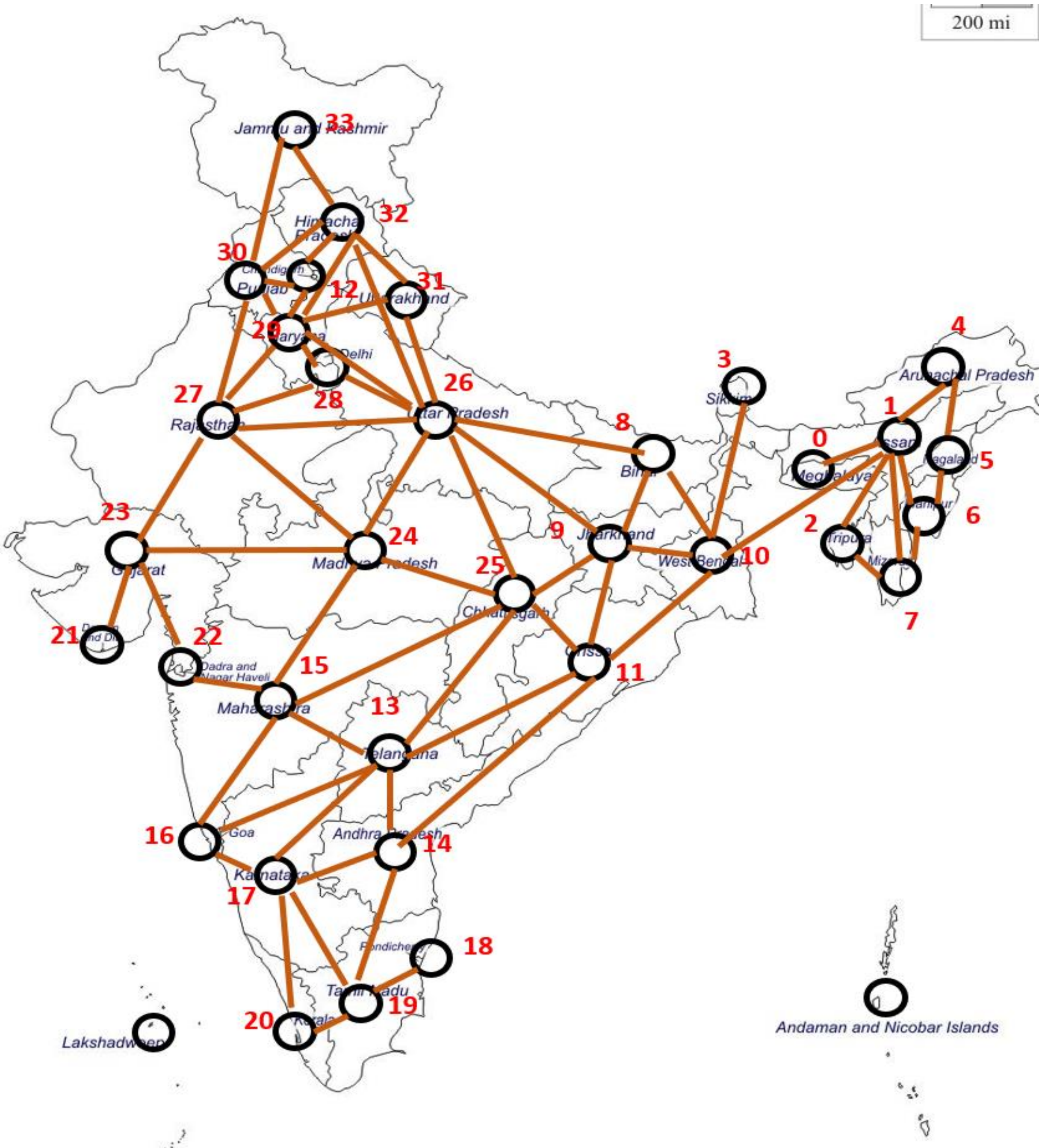
```
while(!cpath.empty()){
    int u = cpath.top();

    if(adj[u].size()==0){
        // if all edges from u are visited
        // pop u and push it to euler path
        epath.push(u);
        cpath.pop();
    }
    else{
        // if all edges from u are not visited
        // select any edge (u, v)
        // push v to current path and remove edge (u, v) from the graph
        cpath.push(adj[u].begin()->first);
        removeEdge(u,adj[u].begin()->first);
    }
}
```

Hình 7.8 Code Hierholzer 2

Theo định hướng R1 = XCdX, thực hiện code trên, ta có kết quả sau: X C d X W Q L A F L K A J K P O J I A E B M E I N M S B Y S T N O U V P Q R F C H D n H j i n m D G B e Y Z T U a b V W c i h m l D k G e f Z a g l k f g h b c d j C R X

CHƯƠNG 8 – CÂU HỎI SỐ 8

8.1. Biểu diễn bản đồ đề cho dưới dạng graph



Hình 8.1. Đồ thị biểu diễn bản đồ

Vì 643 % 4 = 3, nên điểm xuất phát ở Meghalaya (tại điểm xuất phát đánh số 0)

8.2. Ý tưởng để tô màu đồ thị với số màu bé nhất.

Ý tưởng và code dưới đây tham khảo từ [3]:

- Biểu diễn tập hợp các màu dùng để tô dưới dạng mảng các phần tử số 0, 1, 2, …

- Bước 1: Khởi tạo:

+ Dùng danh sách kề để biểu diễn graph

```
private int V; // No. of vertices
private LinkedList<Integer> adj[]; // Adjacency List

// Constructor
Graph(int v) {
    V = v;
    adj = new LinkedList[v];
    for (int i = 0; i < v; ++i)
        adj[i] = new LinkedList();
}

// Function to add an edge into the graph
void addEdge(int v, int w) {
    adj[v].add(w);
    adj[w].add(v); // Graph is undirected
}
```

Hình 8.2. Khởi tạo danh sách cạnh

+ Gán tất cả đỉnh trong graph với giá trị bằng -1 (chưa được tô màu)

```
int result[] = new int[V];

// Initialize all vertices as unassigned
Arrays.fill(result, -1);
```

Hình 8.3. Khởi tạo mảng kết quả

+ Gán giá trị của đỉnh start (ở đây là đỉnh 0) với giá trị là 0 (màu đầu tiên)

```
// Assign the first color to first vertex
result[0] = 0;
```

Hình 8.4. Tô màu đỉnh bắt đầu với màu đầu tiên trong mảng

+ Gán true cho tất cả các phần tử số trong mảng màu nhằm đánh dấu sự khả dụng của tất cả các màu (true là khả dụng, false là ngược lại)

```
boolean available[] = new boolean[V];

// Initially, all colors are available
Arrays.fill(available, true);
```

Hình 8.5. Tạo mảng thể hiện sự khả dụng của từng phần tử màu

- Bước 2: Lặp quá trình sau đến khi chỉ còn lại V-1 đỉnh:

+ Xét đỉnh đang xét với phần tử màu có giá trị bé nhất và khả dụng trong mảng màu hiện có. Nếu tất cả các màu trong mảng đều không khả dụng ta thêm màu mới vào mảng và tô màu mới cho đỉnh đó.

+ Tiếp tục xét đỉnh được đánh số liền sau đỉnh có số vừa mới được tô màu. Và lặp lại bước trên.

```java
// Assign colors to remaining V-1 vertices
for (int u = 1; u < V; u++) {
    // Process all adjacent vertices and flag their colors
    // as unavailable
    Iterator<Integer> it = adj[u].iterator();
    while (it.hasNext()) {
        int i = it.next();
        if (result[i] != -1)
            available[result[i]] = false;
    }

    // Find the first available color
    int cr;
    for (cr = 0; cr < V; cr++) {
        if (available[cr])
            break;
    }

    result[u] = cr; // Assign the found color

    // Reset the values back to true for the next iteration
    Arrays.fill(available, true);
}
```

Hình 8.6. Logic chính xử lí việc tô màu

8.3 Kết quả sau khi tô màu:

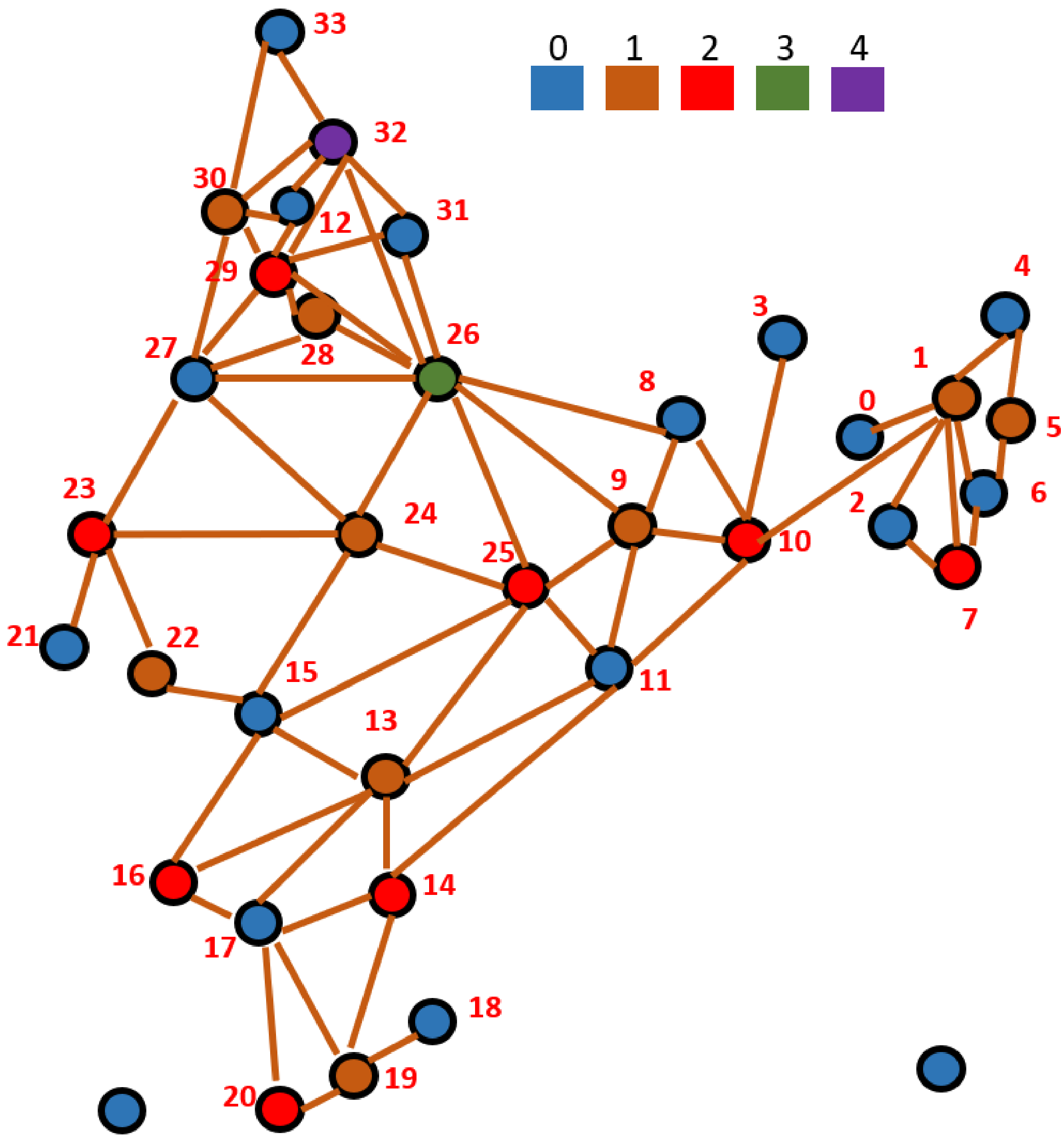
Kết quả của việc thực hiện code trên:

```
Vertex 0 ---> Color 0
Vertex 1 ---> Color 1
Vertex 2 ---> Color 0
Vertex 3 ---> Color 0
Vertex 4 ---> Color 0
Vertex 5 ---> Color 1
Vertex 6 ---> Color 0
Vertex 7 ---> Color 2
Vertex 8 ---> Color 0
Vertex 9 ---> Color 1
Vertex 10 ---> Color 2
Vertex 11 ---> Color 0
Vertex 12 ---> Color 0
Vertex 13 ---> Color 1
Vertex 14 ---> Color 2
Vertex 15 ---> Color 0
Vertex 16 ---> Color 2
Vertex 17 ---> Color 0
Vertex 18 ---> Color 0
Vertex 19 ---> Color 1
Vertex 20 ---> Color 2
Vertex 21 ---> Color 0
Vertex 22 ---> Color 1
Vertex 23 ---> Color 2
Vertex 24 ---> Color 1
Vertex 25 ---> Color 2
Vertex 26 ---> Color 3
Vertex 27 ---> Color 0
Vertex 28 ---> Color 1
Vertex 29 ---> Color 2
Vertex 30 ---> Color 1
Vertex 31 ---> Color 0
Vertex 32 ---> Color 4
Vertex 33 ---> Color 0
```

Hình 8.7 Kết quả qua code

Theo hình 8.7, ta có 5 màu (0, 1, 2, 3, 4) là số màu bé nhất để tô màu bản đồ, với điểm xuất phát tại Meghalaya.

Lưu ý thêm, vì 2 quần đảo Lakshadweep và Andama and Nicobar Islands không liên kết nên ta có thể tô ngay màu 0 cho chúng.



Hình 8.8 Bản đồ sau khi được tô màu

TÀI LIỆU THAM KHẢO

[1] Geeksforgeeks (2017), truy cập tại https://www.youtube.com/watch?v=mHz-mx-8lJ8&t=79s

[2] Slaystudy (2020), truy cập tại https://slaystudy.com/hierholzers-algorithm/?fbclid=IwAR3dYCdIqRyffmvUQffVNZpGjPbM4O-KqiV7g-m9V6_FURM8oIPheU0pt4o

[3] Geeksforgeek (2021), truy cập tại https://www.geeksforgeeks.org/graph-coloring-set-2-greedy-algorithm/