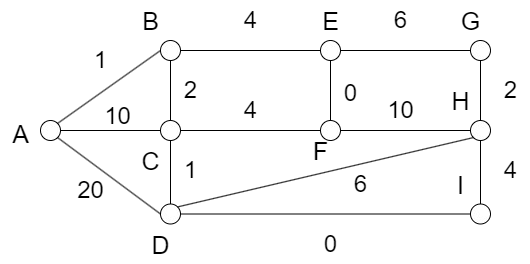
**Bài 1. (2.5 điểm)** Dùng thuật toán **Jarnik – Prim** để trình bày **bảng tiến trình** và **cây khung nhỏ nhất**, giả định rằng thuật toán **luôn ưu tiên đỉnh theo thứ tự bảng chữ cái**. Nút đầu tiên được xác

định bằng **số dư của chữ số cuối MSSV chia cho 3**. Bắt đầu từ đỉnh A nếu số dư là 0, B nếu là 1, và

C nếu là 2. Ví dụ, với MSSV là 01234567, số 7 chia cho 3 dư 1, vậy ta bắt đầu bằng đỉnh B.



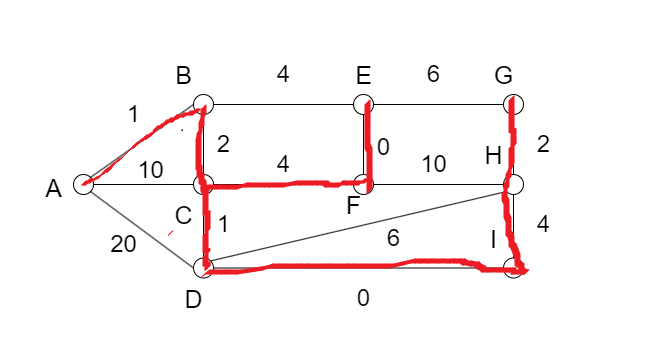
Đi từ C:

W={C,A,B,D,E,F,G,H,I}

R={(C,D),(A,B),(C,B),(D,I),(E,F),(C,F),(G,H),(H,I)}

|  |  |  |
| --- | --- | --- |
| Step | Edge | Weight |
| 0 | (C,D) | 1 |
| 1 | (A,B) | 1 |
| 2 | (B,C) | 2 |
| 3 | (D,I) | 0 |
| 4 | (E,F) | 0 |
| 5 | (C,F) | 4 |
| 6 | (G,H) | 2 |
| 7 | (H,I) | 4 |
|  | SUM: | 14 |

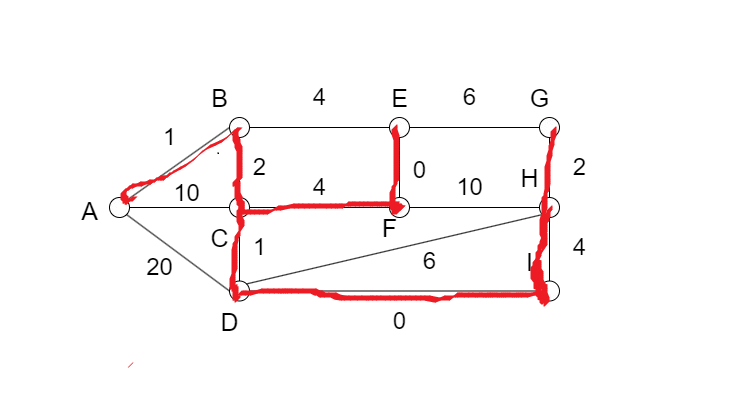
Khung cây nhỏ nhất :



\*ĐI TỪ B:   
W={B,A,C,D,E,F,G,H,I}

R={(B,C),(B,A),(C,D),(D,I),(E,F),(C,F),(G,H),(H,I)}

|  |  |  |
| --- | --- | --- |
| Step | Edge | Weight |
| 0 | (B,C) | 2 |
| 1 | (B,A) | 1 |
| 2 | (C,D) | 1 |
| 3 | (D,I) | 0 |
| 4 | (E,F) | 0 |
| 5 | (C,F) | 4 |
| 6 | (G,H) | 2 |
| 7 | (H,I) | 4 |
|  | SUM: | 14 |



ĐỒ THỊ CÂY KHUNG NHỎ NHẤT:

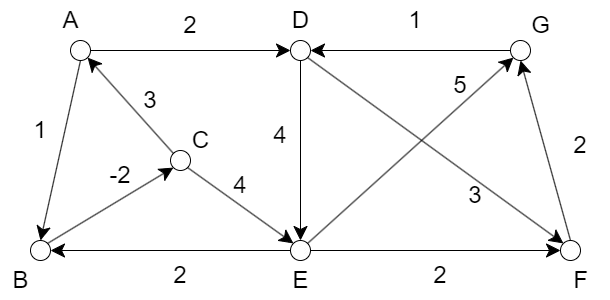
CÂU TỪ A TỰ LÀM

**Bài 2. (2.5 điểm)** Dùng thuật toán **Ford – Bellman** tìm đường đi ngắn nhất tới mọi đỉnh, với thứ tự

các cạnh là **GD, FG, EG, EF, EB, DF, DE, CE, CA, BC, AD, AB.** Nút bắt đầu được xác định bằng

**số dư của chữ số cuối MSSV chia cho 3.** Bắt đầu từ đỉnh A nếu số dư là 0, B nếu là 1, và C nếu là

2. Ví dụ, với MSSV là 01234567, số 7 chia cho 3 dư 1, vậy ta bắt đầu bằng đỉnh B.



Đi từ C : tức là mình sẽ duyệt lần lượt các đỉnh liên quan đến những đỉnh mình đã duyệt lúc đầu và cập nhật giá trị của nó mỗi lần duyệt, nếu nhỏ hơn thì nhận.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | A | B | C | D | E | F | G |  |  |
|  |  |  |  | 0 |  |  |  |  |  |  |
| GD | 1 |  |  |  |  |  |  |  |  |  |
| FG | 2 |  |  |  |  |  |  |  |  |  |
| EG | 5 |  |  |  |  |  |  |  |  |  |
| EF | 2 |  |  |  |  |  |  |  |  |  |
| EB | 2 |  |  |  |  |  |  |  |  |  |
| DF | 3 |  |  |  |  |  |  |  |  |  |
| DE | 4 |  |  |  |  |  |  |  |  |  |
| CE | 4 |  |  |  |  | 4,C |  |  |  |  |
| CA | 3 | 3,C |  |  |  |  |  |  |  |  |
| BC | -2 |  |  |  |  |  |  |  |  |  |
| AD | 2 |  |  |  | 5,A |  |  |  |  |  |
| AB | 1 |  | 4,A |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | A | B | C | D | E | F | G |  |  |
|  |  | 3,C | 4,A | 0,null | 5,A | 4,C |  |  |  |  |
| GD | 1 |  |  |  |  |  |  |  |  |  |
| FG | 2 |  |  |  |  |  |  |  |  |  |
| EG | 5 |  |  |  |  |  |  | 9,E |  |  |
| EF | 2 |  |  |  |  |  | 6,E |  |  |  |
| EB | 2 |  | 4,A |  |  |  |  |  |  |  |
| DF | 3 |  |  |  |  |  |  |  |  |  |
| DE | 4 |  |  |  |  |  |  |  |  |  |
| CE | 4 |  |  |  |  |  |  |  |  |  |
| CA | 3 |  |  |  |  |  |  |  |  |  |
| BC | -2 |  |  |  |  |  |  |  |  |  |
| AD | 2 |  |  |  |  |  |  |  |  |  |
| AB | 1 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

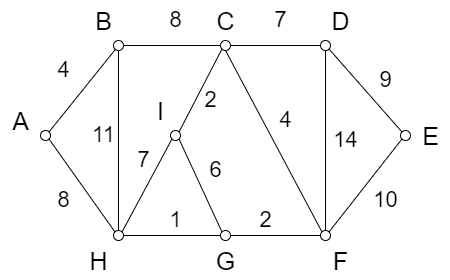
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | A | B | C | D | E | F | G |  |  |
|  |  | 3,C | 4,A | 2,B | 5,A | 4,C | 6,E | 9,E |  |  |
| GD | 1 |  |  |  |  |  |  |  |  |  |
| FG | 2 |  |  |  |  |  |  | 8,F |  |  |
| EG | 5 |  |  |  |  |  |  |  |  |  |
| EF | 2 |  |  |  |  |  |  |  |  |  |
| EB | 2 |  |  |  |  |  |  |  |  |  |
| DF | 3 |  |  |  |  |  |  |  |  |  |
| DE | 4 |  |  |  |  |  |  |  |  |  |
| CE | 4 |  |  |  |  |  |  |  |  |  |
| CA | 3 |  |  |  |  |  |  |  |  |  |
| BC | -2 |  |  |  |  |  |  |  |  |  |
| AD | 2 |  |  |  |  |  |  |  |  |  |
| AB | 1 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | A | B | C | D | E | F | G |  |  |
|  |  | 3,C | 4,A | 2,B | 5,A | 4,C | 6,E | 8,F |  |  |
| GD | 1 |  |  |  | 5,A |  |  |  |  |  |
| FG | 2 |  |  |  |  |  |  | 8,F |  |  |
| EG | 5 |  |  |  |  |  |  |  |  |  |
| EF | 2 |  |  |  |  |  | 6,E |  |  |  |
| EB | 2 |  | 4,A |  |  |  |  |  |  |  |
| DF | 3 |  |  |  |  |  |  |  |  |  |
| DE | 4 |  |  |  |  | 4,C |  |  |  |  |
| CE | 4 |  |  |  |  |  |  |  |  |  |
| CA | 3 | 3,C |  |  |  |  |  |  |  |  |
| BC | -2 |  |  |  |  |  |  |  |  |  |
| AD | 2 |  |  |  |  |  |  |  |  |  |
| AB | 1 |  | 4A |  |  |  |  |  |  |  |
|  |  | 3,C | 4,A | 2,B | 5,A | 4,C | 6,E | 8,F |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| i | A | B | C | D | E | F | G |
| 0 | INF,null | INF,null | 0,null | INF,null | INF,null | INF,null | INF,null |
| 1 | 3,C | ÌNF,null | 0,null | ÌN,null | 4,C | INF,null | INF,null |
| 2 | 3,C | 4,A | 0,null | 5,A | 4,C | 6,E | 9,E |
| 3 | 3,C | 4,A | 0,null | 5,A | 4,C | 6,E | 9,E |
| 4 | 3,C | 4,A | 0,null | 5,A | 4,C | 6,E | 8,F |
| 5 | 3,C | 4,A | 0,null | 5,A | 4,C | 6,E | 8,F |

**Bài 3. (3 điểm)** Dùng thuật toán duyệt theo chiều rộng (**Breadth – first search - BFS**) để duyệt qua toàn bộ đỉnh, giả định rằng thuật toán **luôn ưu tiên đỉnh duyệt tiếp theo theo thứ tự bảng chữ cái**.Nút bắt đầu được xác định bằng **số dư của chữ số cuối MSSV chia cho 3.** Bắt đầu từ đỉnh A nếu số dư là 1, B nếu là 2, và C nếu là 0. Ví dụ, với MSSV là 01234567, số 7 chia cho 3 dư 1, vậy ta bắt

đầu bằng đỉnh .



DFS: chiều sâu

ĐI TỪ A:

(A,B),(B,C),(C,D),(D,E),(E,F),(F,G),(G,H),(H,I)

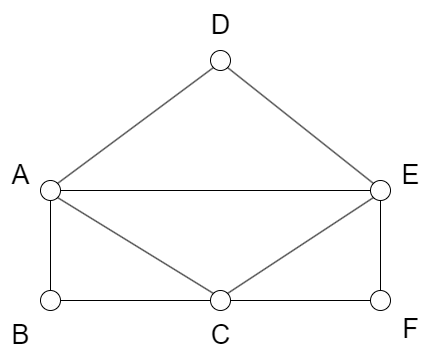
BFS: chiều rộng

ĐI TỪ A:   
(A,B),(A,H),(B,C),(H,G),(H,I),(C,D),(C,F),(D,E)

T NHỚ CÓ CÁCI TRÍCK GÌ MÀ SỐ CẠNH PHẢI NHƯ NÀO VỚI ĐỈNH Ý : BẰNG SỐ ĐỈNH À

**Bài 4**. **(2 điểm)** Dùng thuật toán **Fleury** để tìm đường đi Euler, giả định rằng thuật toán **luôn ưu tiên đỉnh duyệt tiếp theo theo thứ tự bảng chữ cái**. Nút bắt đầu được xác định bằng **số dư của**

**chữ số cuối MSSV chia cho 3.** Bắt đầu từ đỉnh A nếu số dư là 1, B nếu là 2, và C nếu là 0. Ví dụ, với MSSV là 01234567, số 7 chia cho 3 dư 1, vậy ta bắt đầu bằng đỉnh A.



Từ đỉnh B :

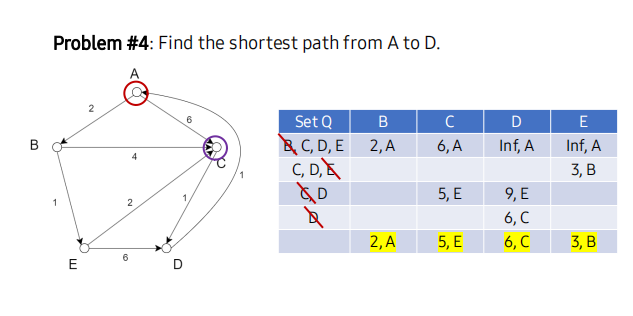
W={B,A,C,E,A,D,E,F,C}

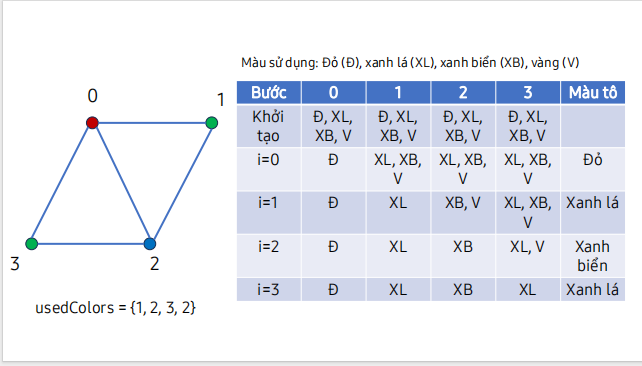
(B,A),(A,C),(C,E),(E,A),(A,D),(D,E),(E,F),(F,C),(C,B)

T={(B,A),(E,F)}

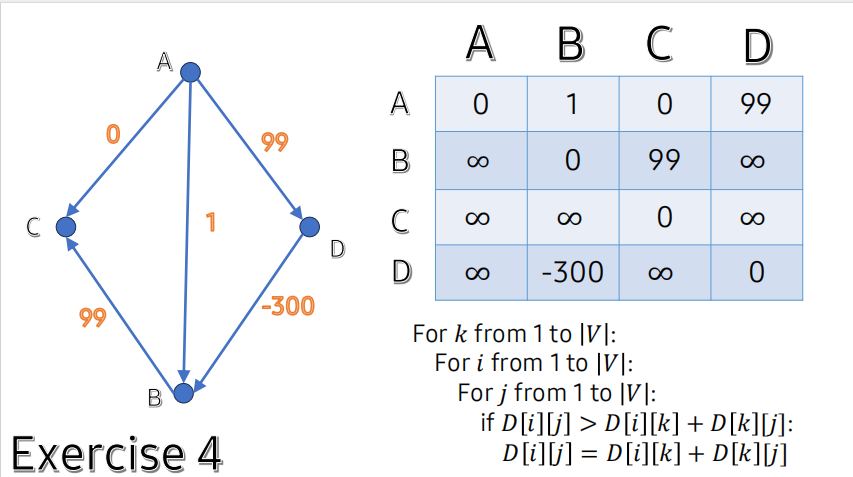
=> CẠNH PHẢI BỎ ĐỂ KHÔNG LẶP ĐỂ TRỞ THÀNH CHU TRÌNH

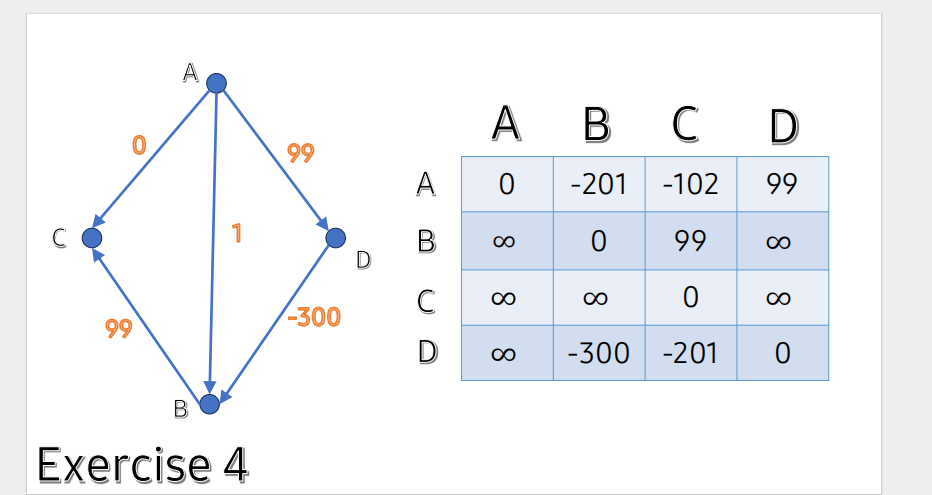
DIJKSTRA:



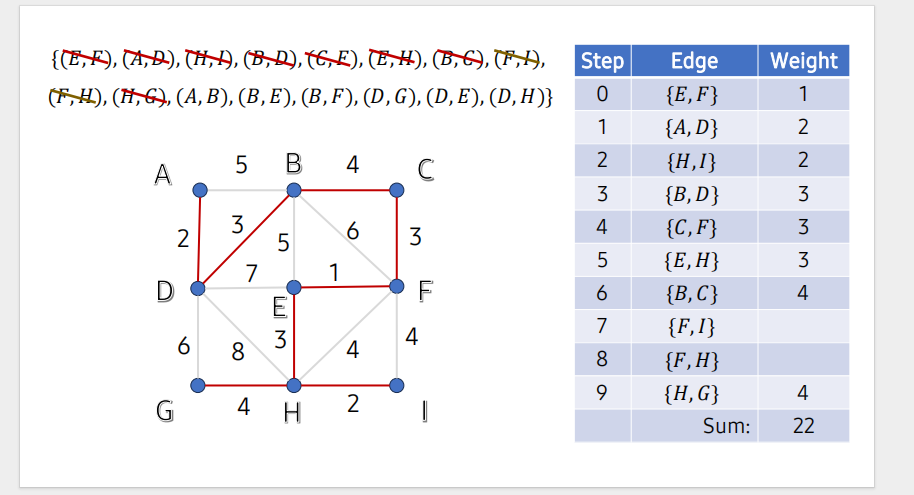
TÔ MÈU:  


FLOYD:





KRUSAL:



***KIỂU MÌNH ĐANG DUYỆT TRỌNG SỐ 4, NÊN MÌNH PHẢI DUYỆT, VÌ ĐI QUA CÁC ĐỈNH RỒI NÊN KHÔNG DUYỆT NỮA.***

**KẾT THÚC KHI SỐ CẠNH = SỐ ĐỈNH - 1**

T chạy hàm for nè :

K=1

I=1

J=1

D[1][1] > D[1][1]+D[1][1]

0 > 0 + 0

K=1

I=1

J=2

....

CỨ VẬY THÔI, RỒI THEO HÀM FOR, THOẢ LÀ NHẬN RỒI THAY THẾ GIÁ TRỊ MỚI VÀO Ô [I][J]

RỒI LẶP LẠI VỚI K++

ỔN RỒI ĐÓ,