BỘ THÔNG TIN TRUYỀN THÔNG

**Học viện Công nghệ Bưu chính Viễn thông**

Khoa CNTT1

*TS. Nguyễn Tất Thắng*

# ĐỀ KIỂM TRA TRẮC NGHIỆM GIỮA KỲ

**Lần 1**

**Môn: Toán rời rạc 2 (midterm2)** *Thời gian làm bài: 110 phút; (30 câu trắc nghiệm)*

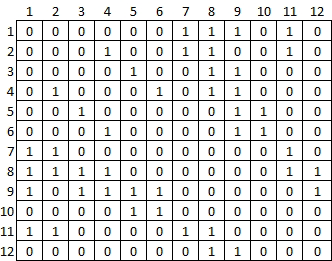
**Mã đề thi 401**

Họ và tên thí sinh:..........................................................................

Mã sinh viên:..................................................................................

Nhóm:.............................................................................................

**Câu 1:** Cho đồ thị vô hướng gồm 12 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



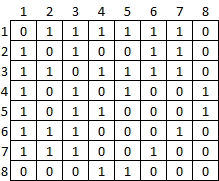
Đường đi từ đỉnh 2 đến đỉnh 11 dựa trên phương pháp duyệt theo chiều sâu DFS là:

**A.** 2 → 4 → 6 → 9 → 1 → 7 → 11 **B.** Không có đáp án nào

## **C.** 2 → 3 → 5 → 6 → 8 → 9 → 11 **D.** 2 → 4 → 3 → 5 → 8 → 9 → 11

**E.** 2 → 3 → 5 → 6 → 8 → 7 → 9 → 11

**Câu 2:** Cho đồ thị vô hướng Euler gồm 8 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Và thuật toán xác định chu trình Euler như sau:

Euler-Cycle(u){

Bước 1: Khởi tạo

stack=rỗng; //khởi tạo stack là rỗng

CE=rỗng; //khởi tạo mảng CE là rỗng

push(stack, u); //đưa đỉnh u vào ngăn xếp Bước 2: Lặp

while(stack != rỗng){

s=get(stack); //lấy đỉnh ở đầu ngăn xếp if(Ke(s) != rỗng){

t=<đỉnh đầu tiên trong Ke(s)>

push (stack, t); //đưa đỉnh t vào ngăn xếp

E=E\{(s, t)}; //loại bỏ cạnh (s, t); Ke(s) = Ke(s) \ {t}

}

else{

s=pop(stack); //loại bỏ s khỏi ngăn xếp

Chuyển s sang mảng CE

}

}

Bước 3: Trả lại kết quả

<lật ngược lại các đỉnh trong CE ta được chu trình Euler>;

}

Áp dụng thuật toán cho đồ thị ở trên (xuất phát từ đỉnh 1), tại một thời điểm nào đó, trạng thái ngăn xếp có chứa danh sách các đỉnh như sau:

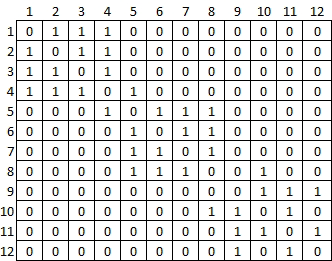
**A.** 1, 2, 3, 1, 4, 3, 5, 1, 6, 2, 7, 3, 2 **B.** 1, 2, 3, 1, 4, 3, 5, 1, 6, 2, 7, 3, 1

**C.** Không có đáp án nào **D.** 1, 2, 3, 1, 4, 3, 5, 1, 6, 2, 7, 3, 6

**E.** 1, 2, 3, 1, 4, 3, 5, 1, 6, 2, 7, 3, 4

1 2 3 1 4 3 5 4 8 5 1 6 2 7 3 6 7 1

**Câu 3:** Cho đồ thị vô hướng gồm 12 đỉnh được biểu diễn dưới dạng ma trận kề như sau:

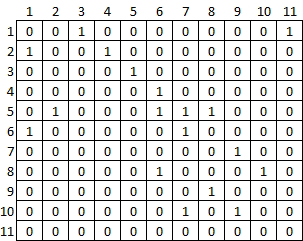


Số các cạnh cầu của đồ thị là:

**A.** 1 **B.** 3 **C.** 4 **D.** 2

**E.** Không có đáp án nào

**Câu 4:** Cho đồ thị có hướng gồm 11 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Bán bậc của các đỉnh của G là:

## **A.** deg+(1)=1, deg+(2)=2, deg+(3)=1, deg+(4)=1, deg+(5)=3, deg+(6)=2, deg+(7)=1, deg+(8)=2,

deg+(9)=1, deg+(10)=2, deg+(11)=1deg-(1)=2, deg-(2)=2, deg-(3)=1, deg-(4)=1, deg-(5)=1, deg-(6)=2, deg-(7)=3, deg-(8)=2,deg-(9)=2, deg-(10)=1, deg-(11)=0

## **B.** deg+(1)=2, deg+(2)=2, deg+(3)=1, deg+(4)=1, deg+(5)=4, deg+(6)=2, deg+(7)=1, deg+(8)=2,

deg+(9)=1, deg+(10)=2, deg+(11)=0 deg-(1)=2, deg-(2)=1, deg-(3)=1, deg-(4)=1, deg-(5)=1, deg-(6)=3, deg-(7)=3, deg-(8)=2,deg-(9)=2, deg-(10)=1, deg-(11)=1

## **C.** deg+(1)=1, deg+(2)=2, deg+(3)=1, deg+(4)=1, deg+(5)=3, deg+(6)=2, deg+(7)=1, deg+(8)=2,

deg+(9)=1, deg+(10)=2, deg+(11)=0deg-(1)=2, deg-(2)=1, deg-(3)=2, deg-(4)=1, deg-(5)=1, deg-(6)=2, deg-(7)=3, deg-(8)=2,deg-(9)=2, deg-(10)=1, deg-(11)=1

**D.** Không có đáp án nào

## **E.** deg+(1)=1, deg+(2)=2, deg+(3)=1, deg+(4)=1, deg+(5)=3, deg+(6)=2, deg+(7)=1, deg+(8)=2,

deg+(9)=2, deg+(10)=2, deg+(11)=1deg-(1)=2, deg-(2)=1, deg-(3)=1, deg-(4)=1, deg-(5)=1, deg-(6)=2, deg-(7)=3, deg-(8)=2,deg-(9)=2, deg-(10)=1, deg-(11)=1

**Câu 5:** Cho thuật toán duyệt đồ thị theo chiều rộng BFS sử dụng thuật toán lặp và sử dụng hàng đợi như sau:

1 BFS(u){

Bước 1: Khởi tạo

1. queue = rỗng;
2. push(queue, u);
3. chuaxet[u]=false; //xác nhận u được duyệt Bước 2: Lặp
4. while(queue != rỗng){
5. s = front(queue); pop(queue); //lấy và xóa 1 đỉnh ở đầu hàng đợi
6. <Thăm đỉnh s>;
7. Lấy 1 đỉnh t thuộc Ke(s){
8. if(chuaxet[t]){
9. push(queue, t);
10. chuaxet[t]=false; //xác nhận t được duyệt 13 }

14 }

15 }

Bước 3: Trả lại kết quả

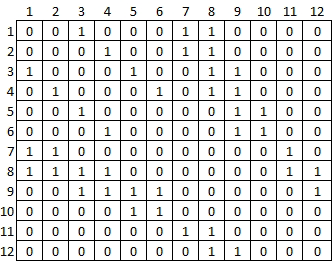
16 return <tập đỉnh đã duyệt>; 17 }

Dòng sai là:

**A.** 4 **B.** 11 **C.** 6 **D.** Không có đáp án nào

**E.** 9

**Câu 6:** Cho đồ thị vô hướng gồm 12 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Đường đi từ đỉnh 1 đến đỉnh 12 dựa trên phương pháp duyệt theo chiều rộng BFS là:

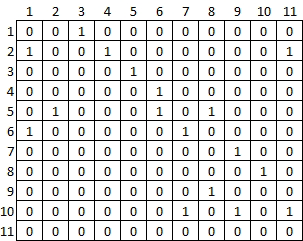
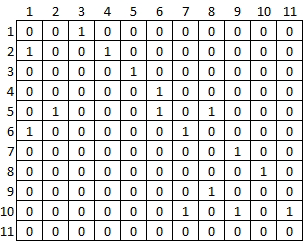
**A.** 1 → 4 → 12 **B.** 1 → 9 → 12 **C.** Không có đáp án nào **D.** 1 → 8 → 12

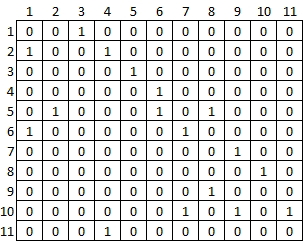
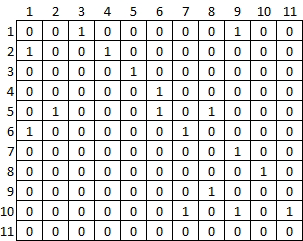
## **E.** 1 → 8 → 10 → 12

**Câu 7:** Cho đồ thị có hướng G = <V, E> gồm 11 đỉnh được biểu diễn dưới dạng danh sách kề như sau: Ke(1) = {3} Ke(2) = {1, 4} Ke(3) = {5} Ke(4) = {6} Ke(5) = {2, 6, 8} Ke(6) = {1, 7}

Ke(7) = {9} Ke(8) = {10} Ke(9) = {8} Ke(10) = {7, 9, 11} Ke(11)={}

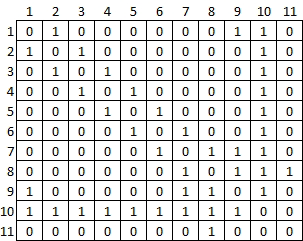
Ma trận kề của đồ thị G như sau:

**A. ** **B. **

**C. D.**

**E.** Không có đáp án nào

**Câu 8:** Cho đồ thị vô hướng gồm 11 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Bậc của các đỉnh của đồ thị là:

**A.** deg(1)=deg(2)=deg(3)=deg(4)=deg(5)=deg(6)=deg(7)=deg(8)=3, deg(9)=4, deg(10)=9, deg(11)=1

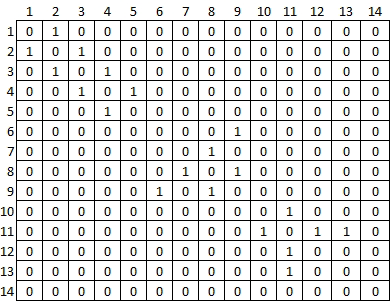
**B.** deg(1)=deg(2)=deg(3)=deg(4)=deg(5)=3, deg(6)=deg(7)=deg(8)=deg(9)=4, deg(10)=9, deg(11)=2

**C.** deg(1)=deg(2)=deg(3)=deg(4)=deg(5)=deg(6)=deg(7)=deg(8)=deg(9)=3, deg(10)=10, deg(11)=1

**D.** deg(1)=deg(2)=deg(3)=deg(4)=deg(5)=deg(6)=3, deg(7)=deg(8)=deg(9)=4, deg(10)=9, deg(11)=1

**E.** Không có đáp án nào

**Câu 9:** Cho đồ thị vô hướng gồm 14 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Số thành phần liên thông của đồ thị đã cho là:

**A.** 4 **B.** 5 **C.** 2 **D.** 3

**E.** Không có đáp án nào

**Câu 10:** Cho thuật toán duyệt số thành phần liên thông và một đồ thị vô hướng có ma trận kề như dưới đây:

1. Duyet-TPLT(){ // duyệt thành phần liên thông Bước 1: Khởi tạo
2. soTPTL = 0; // khởi tạo số thành phần liên thông = 0 Bước 2: Lặp

|  |  |  |
| --- | --- | --- |
| 3 | for(u thuộc V){ | // lặp trên tập đỉnh |
| 4 | if(chuaxet[u]){ |  |
| 5 | soTPTL = soTPTL + 1; | // ghi nhận số TPLT |
| 6 | BFS(u); | // có thể gọi DFS(u) |

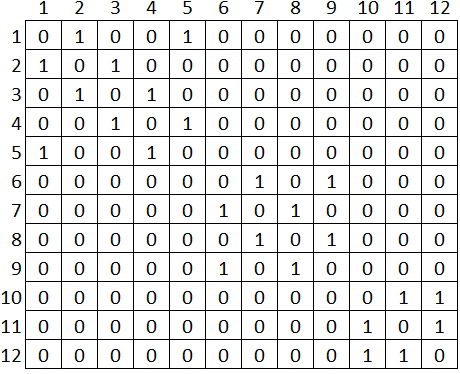
7 <Ghi nhận các đỉnh thuộc TPLT>;

}

9 }

Bước 3: Trả lại kết quả

10 return <các TPLT>; 11 }

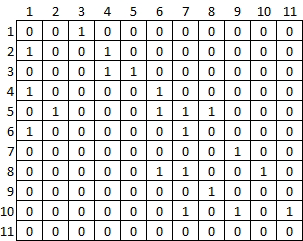
Ma trận kề của đồ thị vô hướng có 12 đỉnh:

Khi thực hiện thuật toán với đồ thị đã cho, với u = 8 tại dòng 3, kết quả ghi nhận tại dòng 7 là:

**A.** 4, 5, 6, 7 **B.** Không có đáp án nào **C.** 7, 8, 9, 10 **D.** 5, 6, 7, 8

**E.** 6, 7, 8, 9

**Câu 11:** Cho đồ thị có hướng gồm 11 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Danh sách kề của các đỉnh của G là:

**A.** Không có đáp án nào

**B.** Ke(1) = {3} Ke(2) = {1, 4} Ke(3) = {4, 5} Ke(4) = {1, 6} Ke(5) = {2, 6, 7, 8} Ke(6)

= {1, 7} Ke(7) = {9} Ke(8) = {6, 7, 10} Ke(9) = {8} Ke(10) = {7, 9, 11} Ke(11) = {}

**C.** Ke(1) = {3} Ke(2) = {1, 4} Ke(3) = {4, 5} Ke(4) = {6} Ke(5) = {2, 6, 8} Ke(6) =

{1, 7} Ke(7) = {9} Ke(8) = {6, 7, 10} Ke(9) = {8} Ke(10) = {7, 9, 11} Ke(11) = {2}

**D.** Ke(1) = {3} Ke(2) = {1, 4} Ke(3) = {4, 5} Ke(4) = {6} Ke(5) = {2, 6, 7, 8}

Ke(6) = {1, 7} Ke(7) = {9} Ke(8) = {6, 7, 10} Ke(9) = {8} Ke(10) = {7, 9, 10} Ke(11) =

{1}

**E.** Ke(1) = {3} Ke(2) = {1, 4} Ke(3) = {4, 5} Ke(4) = {6} Ke(5) = {2, 7, 8} Ke(6) =

{1, 7} Ke(7) = {9} Ke(8) = {6, 7, 10} Ke(9) = {8} Ke(10) = {7, 9, 11} Ke(11) = {10}

**Câu 12:** Cho thuật toán duyệt đồ thị theo chiều sâu DFS sử dụng thuật toán lặp và sử dụng ngăn xếp như sau:

1. DFS(u){

Bước 1: Khởi tạo

1. stack=rỗng; //khởi tạo stack là rỗng
2. push(stack, u); //đưa đỉnh u vào stack
3. <Thăm đỉnh u>; //duyệt đỉnh u
4. chuaxet[u]=false; //xác nhận đã duyệt u Bước 2: Lặp
5. while(stack != rỗng){
6. s = get(stack); pop(stack); //lấy và xóa 1 đỉnh ở đầu stack

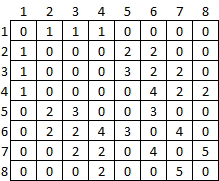
|  |  |  |
| --- | --- | --- |
| 8 | for(t thuộc Ke(s)){ |  |
| 9 | if(chuaxet[t]){ | //nếu chưa duyệt t |
| 10 | <Thăm đỉnh t>; | //duyệt đỉnh t |
| 11 | chuaxet[t]; | //xác nhận t đã được duyệt |
| 12 | push(stack, t); | //đưa t vào stack |
| 13 | push(stack, s); | //đưa s vào stack |
| 14 | break; | //chỉ lấy một đỉnh t |
| 15 | } |  |
| 16 | } |  |
| 17 | } |  |
|  | Bước 3: Trả lại kết quả |  |
| 18 | return <tập đỉnh đã duyệt>; |  |
| 19 } |  |  |

Dòng sai là:

**A.** 9, 10 **B.** 2, 3 **C.** 12, 13 **D.** Không có đáp án nào

**E.** 5, 6

**Câu 13:** Cho đồ thị vô hướng liên thông gồm 12 đỉnh được biểu diễn dưới dạng ma trận trọng số như sau:



Áp dụng thuật toán Kruskal, cho biết độ dài cây và danh sách cạnh của cây bao trùm bé nhất:

**A.** dH = 11; (1, 2), (1, 3), (1, 4), (2, 5), (2, 6), (4, 7), (4, 8) **B.** dH = 11; (1, 2), (1, 3), (1, 4), (2, 5), (3, 6),

(3, 7), (4, 8)

**C.** dH = 11; (1, 2), (1, 3), (1, 4), (2, 5), (3, 6), (4, 7), (4, 8) **D.** dH = 11; (1, 2), (1, 3), (1, 4), (2, 5), (2,

6), (3, 7), (4, 8)

**E.** Không có đáp án nào

**Câu 14:** Cho đồ thị vô hướng G = <V, E> gồm 11 đỉnh được biểu diễn dưới dạng danh sách kề như sau: Ke(1) = {2, 9, 10, 11} Ke(2) = {1, 3, 10} Ke(3) = {2, 4, 10} Ke(4) = {3, 5, 10}

Ke(5) = {4, 6, 10} Ke(6) = {5, 7, 10, 11} Ke(7) = {6, 8, 9} Ke(8) = {7, 9}

Ke(9) = {1, 7, 8} Ke(10) = {1, 2, 3, 4, 5, 6} Ke(11) = {1, 6}

Bậc của mỗi đỉnh trên đồ thị như sau:

**A.** Không có đáp án nào

**B.** deg(1)=deg(6)=4, deg(2)=deg(3)=deg(4)=deg(5)=deg(7)=deg(9)=3, deg(8)=deg(11)=2, deg(10)=6

**C.** deg(1)=deg(2)=deg(3)=deg(4)=deg(5)=deg(6)=deg(7)=deg(9)=3, deg(8)=2, deg(10)=7, deg(11)=4

**D.** deg(1)=deg(2)=deg(3)=deg(4)=deg(5)=deg(6)=3, deg(7)=deg(8)=deg(9)=4, deg(10)=6, deg(11)=1

**E.** deg(1)=deg(2)=deg(3)=deg(4)=deg(5)=deg(6)=3, deg(7)=deg(8)=deg(9)=2, deg(10)=5, deg(11)=3

**Câu 15:** Cho đồ thị có hướng G = <V, E> gồm 11 đỉnh được biểu diễn dưới dạng danh sách kề như sau: Ke(1) = {3} Ke(2) = {1, 4} Ke(3) = {5} Ke(4) = {6} Ke(5) = {2} Ke(6) = {7}

Ke(7) = {9} Ke(8) = {10} Ke(9) = {2, 8} Ke(10) = {9, 11} Ke(11) = {}

Bán bậc của mỗi đỉnh trên đồ thị như sau:

**A.** deg+(1)=1, deg+(2)=2, deg+(3)=1, deg+(4)=1, deg+(5)=1, deg+(6)=1, deg+(7)=1, deg+(8)=1, deg+(9)=1,

deg+(10)=1, deg+(11)=2 deg-(1)=1, deg-(2)=1, deg-(3)=1, deg-(4)=2, deg-(5)=1, deg-(6)=1, deg-(7)=1, deg- (8)=1, deg-(9)=2, deg-(10)=1, deg-(11)=1

**B.** deg+(1)=1, deg+(2)=1, deg+(3)=1, deg+(4)=1, deg+(5)=1, deg+(6)=1, deg+(7)=1, deg+(8)=1, deg+(9)=1,

deg+(10)=1, deg+(11)=1 deg-(1)=1, deg-(2)=1, deg-(3)=1, deg-(4)=1, deg-(5)=1, deg-(6)=1, deg-(7)=1, deg- (8)=1, deg-(9)=2, deg-(10)=1, deg-(11)=0

**C.** deg+(1)=1, deg+(2)=2, deg+(3)=1, deg+(4)=1, deg+(5)=1, deg+(6)=1, deg+(7)=1, deg+(8)=1, deg+(9)=2,

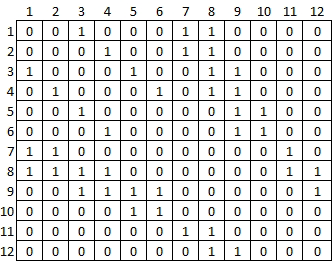
deg+(10)=1, deg+(11)=0 deg-(1)=1, deg-(2)=2, deg-(3)=1, deg-(4)=1, deg-(5)=1, deg-(6)=1, deg-(7)=1, deg- (8)=1, deg-(9)=2, deg-(10)=1, deg-(11)=1

**D.** deg+(1)=1, deg+(2)=2, deg+(3)=1, deg+(4)=1, deg+(5)=1, deg+(6)=1, deg+(7)=1, deg+(8)=1, deg+(9)=1,

deg+(10)=1, deg+(11)=1 deg-(1)=1, deg-(2)=1, deg-(3)=2, deg-(4)=1, deg-(5)=1, deg-(6)=1, deg-(7)=1, deg- (8)=1, deg-(9)=2, deg-(10)=1, deg-(11)=1

**E**. Không có đáp án nào

**Câu 16:** Cho đồ thị vô hướng gồm 12 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Đường đi từ đỉnh 1 đến đỉnh 12 dựa trên phương pháp duyệt theo chiều sâu DFS là:

## **A.** 1 → 7 → 6 → 5 → 11 → 10 → 9 → 12 **B.** 1 → 4 → 7 → 5 → 6 → 11 → 9 → 10 → 12

**C.** 1 → 3 → 5 → 9 → 4 → 2 → 7 → 11 → 8 → 12 **D.** 1 → 3 → 4 → 5 → 7 → 6 → 11 → 10

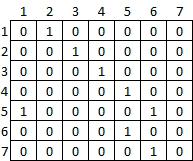
## → 9 → 12

**E.** Không có đáp án nào

**Câu 17:** Cho thuật toán dưới đây xác định tính liên thông mạnh của đồ thị có hướng:

|  |  |  |
| --- | --- | --- |
| 1 | bool Strongly\_Connected(G=<V,E>){ | // kiểm tra tính liên thông mạnh của đồ thị có hướng G |
| 2 | ReInit(); | // Với mọi đỉnh u thuộc tập đỉnh V: chuaxet[u]=true; |
| 3 | for(u thuộc V){ | // lặp trên tập đỉnh V |
| 4 | if(DFS(u) != V) | // có thể kiểm tra BFS(u) != V |
| 5 | return false; | // đồ thị không liên thông mạnh |
| 6 | else |  |
| 7 | ReInit(); | // khởi tạo lại mảng chuaxet[] |
| 8 | } |  |
| 9 | return true; | // đồ thị liên thông mạnh |
| 10 | } |  |

Và cho đồ thị có hướng gồm 7 đỉnh cho bởi ma trận kề như dưới đây:



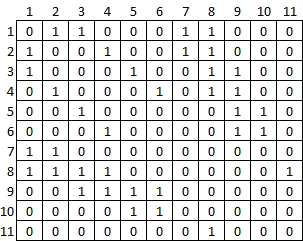
Cho biết thuật toán trên sẽ thực hiện từ đỉnh 1 và những đỉnh nào thì kết thúc; cho biết tính liên thông mạnh (hay yếu) của đồ thị:

**A.** 1, 2, 3, 4, 5, 6, 7; liên thông mạnh **B.** 1; liên thông yếu

**C.** 1, 2, 3, 4, 5, 6, 7; liên thông yếu **D.** Không có đáp án nào

**E.** 1, 2, 3, 4; liên thông yếu

**Câu 18:** Cho đồ thị vô hướng gồm 11 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Kết quả duyệt theo chiều sâu từ đỉnh số 1 - DFS(1) là:

**A.** 1 3 5 9 4 2 7 6 8 10 11 **B.** 1 5 3 9 4 2 7 8 6 10 11

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **C.** 1 | 3 | 5 | 9 | 2 | 4 | 7 | 8 | 11 | 6 | 10 | **D.** Không có đáp án nào |
| **E.** 1 | 2 | 4 | 6 | 9 | 3 | 5 | 10 | 8 | 11 | 7 |  |

**Câu 19:** Cho thuật toán duyệt đồ thị theo chiều sâu DFS sử dụng thuật toán duyệt đệ quy như sau:

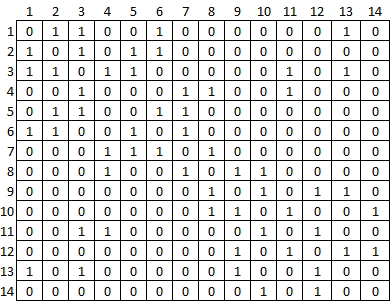
|  |  |  |
| --- | --- | --- |
| 1 | DFS(u){ | //u là đỉnh bắt đầu duyệt |
| 2 | <Thăm đỉnh u>; | //duyệt đỉnh u |
| 3 | Đánh dấu đã thăm u; | //xác nhận đỉnh u đã duyệt |
| 4 | Lấy 1 đỉnh v thuộc Ke(u){ |  |
| 5 | if(chưa thăm v) | //nếu v chưa được duyệt |
| 6 | DFS(v); | //duyệt theo chiều sâu từ đỉnh v |
| 7 | } |  |
| 8 | } |  |

Dòng sai là:

**A.** 1 **B.** 4 **C.** 3 **D.** Không có đáp án nào

**E.** 5

**Câu 20:** Cho đồ thị vô hướng gồm 14 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Chu trình Euler xuất phát từ đỉnh 6 là:

6 1 2 3 1 13 3 4 7 5 2 6 5 3 11 4 8 9 10 11 12 9 13 12 14 10 8 7 6

**A.** 6 → 1 → 2 → 3 → 1 → 13 → 3 → 4 → 7 → 5 → 3 → 11 → 4 → 8 → 9 → 13 → 14 → 12 →

11 → 10 → 7 → 8 → 6

**B.** 6 → 1 → 2 → 3 → 1 → 13 → 4 → 3 → 7 → 5 → 3 → 11 → 4 → 8 → 9 → 13 → 12 → 11 →

10 → 14 → 8 → 7 → 6

**C.** 6 → 1 → 2 → 3 → 1 → 13 → 3 → 4 → 7 → 5 → 2 → 6 → 5 → 3 → 11 → 4 → 8 → 9 → 10 →

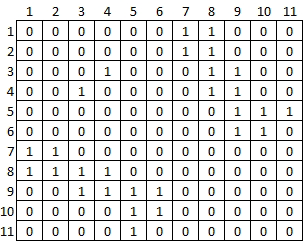
11 → 12 → 9 → 13 → 12 → 14 → 10 → 8 → 7 → 6

**D.** 6 → 1 → 3 → 2 → 1 → 13 → 3 → 4 → 7 → 5 → 3 → 11 → 4 → 8 → 9 → 13 → 12 → 11 →

10 → 8 → 7 → 14 → 6

**E.** Không có đáp án nào

**Câu 21:** Cho đồ thị vô hướng gồm 11 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Kết quả duyệt theo chiều rộng từ đỉnh số 1 - BFS(1) là:

**A.** Không có đáp án nào **B.** 1 7 8 2 3 4 9 6 5 10 11 **C.** 1 7 8 2 3 4 5 9 6 11 10 **D.** 1 7 8 2 3 4 9 5 6 10 11 **E.** 1 7 8 3 2 4 9 5 6 11 10

**Câu 22:** Cho thuật toán duyệt đồ thị theo chiều sâu DFS sử dụng thuật toán lặp và ngăn xếp như sau:

1. DFS(u){

Bước 1: Khởi tạo

1. stack = rỗng; //khởi tạo stack rỗng
2. push(stack, u); //đưa đỉnh u vào stack
3. <Thăm đỉnh u>; //duyệt đỉnh u
4. chuaxet[u]=false; //xác nhận đã duyệt u Bước 2: Lặp
5. while(stack != rỗng){
6. s=top(stack); //lấy 1 đỉnh ở đầu stack
7. for(t thuộc Ke(s)){
8. if(chuaxet[t]){ //nếu chưa duyệt t
9. <Thăm đỉnh t>; //duyệt đỉnh t
10. chuaxet[t]=false; //t đã được duyệt
11. push(stack, s); //đưa s vào stack
12. push(stack, t); //đưa t vào stack
13. break; //chỉ lấy một đỉnh t 15 }

16 }

17 }

Bước 3: Trả lại kết quả

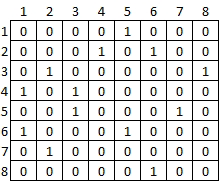
18 return <tập đỉnh đã duyệt>; 19 }

Dòng sai là:

**A.** 7 **B.** Không có đáp án nào **C.** 3 **D.** 6

**E.** 9

**Câu 23:** Cho đồ thị có hướng nửa Euler gồm 8 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Và thuật toán xác định chu trình Euler như sau:

Euler-Cycle(u){

Bước 1: Khởi tạo

stack=rỗng; //khởi tạo stack là rỗng

CE=rỗng; //khởi tạo mảng CE là rỗng

push(stack, u); //đưa đỉnh u vào ngăn xếp Bước 2: Lặp

while(stack != rỗng){

s=get(stack); //lấy đỉnh ở đầu ngăn xếp if(Ke(s) != rỗng){

t=<đỉnh đầu tiên trong Ke(s)>

push (stack, t); //đưa đỉnh t vào ngăn xếp

E=E\{(s, t)}; //loại bỏ cạnh (s, t); Ke(s) = Ke(s) \ {t}

}

else{

}

}

s=pop(stack); //loại bỏ s khỏi ngăn xếp Chuyển s sang mảng CE

Bước 3: Trả lại kết quả

<lật ngược lại các đỉnh trong CE ta được chu trình Euler>;

}

Áp dụng thuật toán cho đồ thị ở trên (xuất phát từ đỉnh 4), tại một thời điểm nào đó, trạng thái ngăn xếp có chứa danh sách các đỉnh như sau:

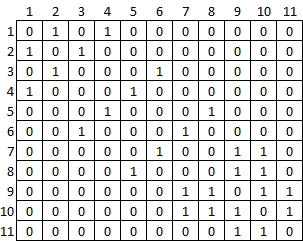
4 1 5 3 2 4 3 8 6 5 7 2 6 1

**A.** 4, 1, 5, 3, 2, 4, 3, 8, 6, 5, 7, 2, 6 **B.** 4, 1, 5, 3, 2, 4, 5, 8, 6, 5, 7, 2, 6

**C.** Không có đáp án nào **D.** 4, 1, 5, 3, 2, 4, 3, 8, 2, 5, 7, 2, 6

**E.** 4, 1, 5, 3, 2, 4, 3, 8, 6, 5, 7, 2, 1

**Câu 24:** Cho đồ thị vô hướng gồm 10 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Danh sách kề của các đỉnh là:

**A.** Ke(1) = {4} Ke(2) = {3} Ke(3) = {2, 6} Ke(4) = {1, 5, 9} Ke(5) = {4, 8}

Ke(6) = {3, 7} Ke(7) = {6, 9, 10} Ke(8) = {5, 9, 10} Ke(9) = {7, 8, 10} Ke(10) = {7, 8, 9}

Ke(11) = {8, 9}

**B.** Không có đáp án nào

**C.** Ke(1) = {4} Ke(2) = {3} Ke(3) = {2, 6} Ke(4) = {1, 5, 8} Ke(5) = {4, 8}

Ke(6) = {3, 7} Ke(7) = {6, 9, 10} Ke(8) = {5, 9, 10} Ke(9) = {7, 8, 10} Ke(10) = {7, 8, 9}

Ke(11) = {10}

**D.** Ke(1) = {2, 4} Ke(2) = {1, 3} Ke(3) = {2, 6} Ke(4) = {1, 5, 7} Ke(5) = {4, 8}

Ke(6) = {3, 7} Ke(7) = {6, 9, 10} Ke(8) = {5, 9, 10} Ke(9) = {7, 8, 10} Ke(10) = {7, 8, 9}

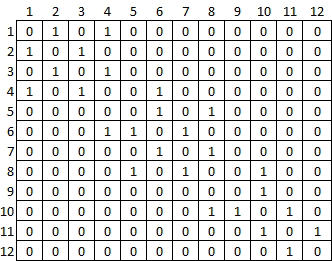
Ke(11) = {9}

**E.** Ke(1) = {2, 4} Ke(2) = {1, 3} Ke(3) = {2, 6} Ke(4) = {1, 5} Ke(5) = {4, 8}

Ke(6) = {3, 7} Ke(7) = {6, 9, 10} Ke(8) = {5, 9, 10} Ke(9) = {7, 8, 10, 11} Ke(10) = {7, 8,

9, 11} Ke(11) = {9, 10}

**Câu 25:** Cho đồ thị vô hướng gồm 12 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Số các đỉnh trụ của đồ thị là:

**A.** Không có đáp án nào **B.** 7 **C.** 5 **D.** 4

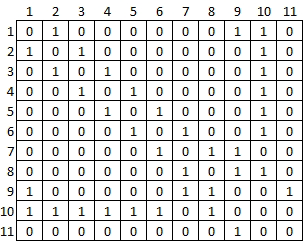
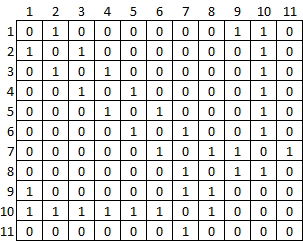
**E.** 6

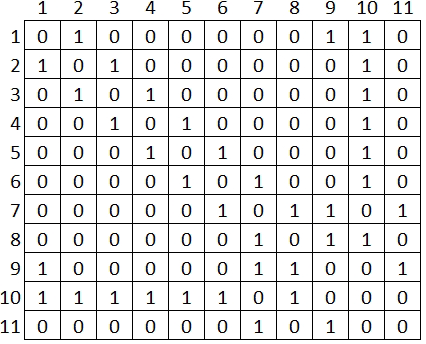
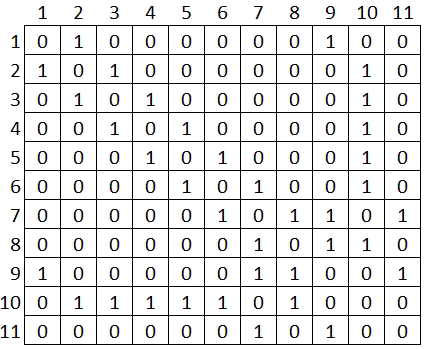
**Câu 26:** Cho đồ thị vô hướng G = <V, E> gồm 11 đỉnh được biểu diễn dưới dạng danh sách kề như sau: Ke(1) = {2, 9, 10} Ke(2) = {1, 3, 10} Ke(3) = {2, 4, 10} Ke(4) = {3, 5, 10}

Ke(5) = {4, 6, 10} Ke(6) = {5, 7, 10} Ke(7) = {6, 8, 9, 11} Ke(8) = {7, 9, 10}

Ke(9) = {1, 7, 8, 11} Ke(10) = {1, 2, 3, 4, 5, 6, 8} Ke(11) = {7, 9}

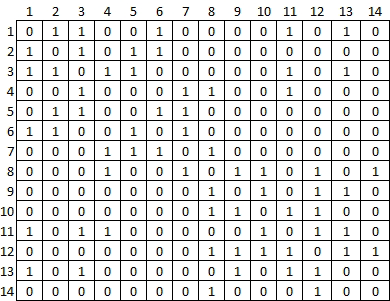
Biểu diễn G dưới dạng ma trận kề như sau:

**A. ** **B. **

**C. ** **D. **

**E.** Không có đáp án nào

**Câu 27:** Cho đồ thị vô hướng gồm 14 đỉnh được biểu diễn dưới dạng ma trận kề như sau:



Đường đi Euler xuất phát từ đỉnh 1 đến đỉnh 13 là:

**A.** 1 → 2 → 3 → 1 → 6 → 2 → 5 → 3 → 4 → 7 → 5 → 6 → 7 → 8 → 4 → 11 → 1 → 13 → 3 → 11

→ 10 → 8 → 9 → 10 → 12 → 8 → 14 → 12 → 9 → 13 → 11 → 12 → 13

**B.** 1 → 2 → 3 → 1 → 6 → 4 → 7 → 3 → 5 → 7 → 8 → 4 → 11 → 1 → 13 → 3 → 11 → 10 → 8 →

9 → 13 → 11 → 12 → 13 → 12 → 8 → 14 → 12 → 9 → 13 → 11 → 12 → 13

**C.** Không có đáp án nào

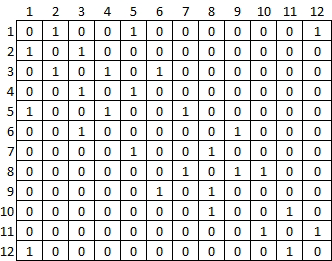
**D.** 1 → 2 → 3 → 1 → 6 → 7 → 4 → 3 → 5 → 7 → 8 → 4 → 11 → 1 → 13 → 3 → 11 → 10 → 8 →

9 → 13 → 12 → 11 → 12 → 8 → 14 → 12 → 9 → 13 → 11 → 12 → 13

**E.** 1 → 2 → 3 → 1 → 6 → 7 → 4 → 3 → 5 → 8 → 7 → 4 → 11 → 1 → 13 → 3 → 11 → 10 → 8 →

9 → 13 → 11 → 14 → 12 → 8 → 14 → 12 → 9 → 13 → 11 → 12 → 13

**Câu 28:** Cho đồ thị vô hướng gồm 12 đỉnh được biểu diễn dưới dạng ma trận kề như sau:

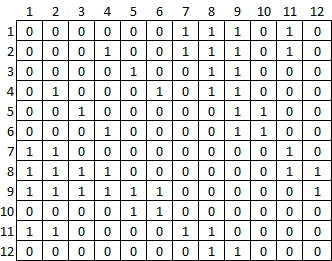


Số thành phần liên thông của đồ thị đã cho là:

**A.** 1 **B.** 3 **C.** Không có đáp án nào **D.** 2

**E.** 4

**Câu 29:** Cho đồ thị vô hướng gồm 12 đỉnh được biểu diễn dưới dạng ma trận kề như sau:

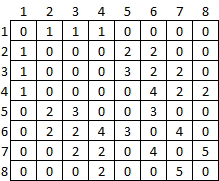


Đường đi từ đỉnh 2 đến đỉnh 11 dựa trên phương pháp duyệt theo chiều rộng BFS là:

**A.** 2 → 8 → 9 → 11 **B.** 2 → 9 → 10 **C.** 2 → 3 → 6 → 11 **D.** Không có đáp án nào

**E.** 2 → 11

**Câu 30:** Cho đồ thị vô hướng liên thông gồm 12 đỉnh được biểu diễn dưới dạng ma trận trọng số như sau:



Áp dụng thuật toán Prim từ đỉnh 8, cho biết độ dài cây và danh sách cạnh của cây bao trùm bé nhất:

**A.** dH = 11; (4, 8), (1, 2), (1, 3), (1, 4), (2, 5), (3, 6), (3, 7) **B.** dH = 11; (4, 8), (1, 2), (1, 3), (1, 4), (2, 5),

(3, 6), (4, 7)

**C.** Không có đáp án nào **D.** dH = 11; (4, 8), (1, 2), (1, 3), (1, 4), (2, 5), (2, 6),

(4, 7)

**E.** dH = 11; (4, 8), (1, 4), (1, 2), (1, 3), (2, 5), (2, 6), (3, 7)