***Form 3a***

|  |  |
| --- | --- |
| POSTS AND TELECOMMUNICATIONS INSTITUTE OF TECHNOLOGY  FACULTY**:**  INFORMATION TECHNOLOGY I  **DEPARTMENT:** COMPUTER SCIENCE | **FINAL EXAM**  **(written exam)** |
| **Course: Discrete Mathematics 2 (Semester 2, Academic year 2022-2023)** | | |
| **Classes: Honors-Program Classes** | **Time:** 90 minutes | |

**Exam Paper**: 3

**Question 1** (2 marks)

Given a directed graph G = <V, E> having 10 vertices. G is represented by the adjacency list Adj as follows:

Adj(1) = {2, 9} Adj(6) = {3, 5, 7, 8, 10}

Adj(2) = {3, 4, 5, 6} Adj(7) = {8, 9}

Adj(3) = {4, 5} Adj(8) = {1, 9}

Adj(4) = {6} Adj(9) = {10}

Adj(5) = {1, 4, 7} Adj(10) = {1, 2, 3}

1. Determine the indegree and outdegree of the vertices of G.
2. Represent G using an adjacency matrix.

*Cho đồ thị có hướng G = <V, E> gồm 10 đỉnh được biểu diễn dưới dạng danh sách kề như sau:*

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

*Ke(2) = {3, 4, 5, 6} Ke(7) = {8, 9}*

*Ke(3) = {4, 5} Ke(8) = {1, 9}*

*Ke(4) = {6} Ke(9) = {10}*

*Ke(5) = {1, 4, 7} Ke(10) = {1, 2, 3}*

*a) Tìm bán bậc vào và bán bậc ra của mỗi đỉnh trên đồ thị.*

*b) Biểu diễn đồ thị G dưới dạng ma trận kề.*

**Question 2** (2 marks)

a) Write a function called DFS(int u) in C/C++ based on the stack data structure. The DFS(int u) function does the depth first search of a graph G = <V, E>. The search starts from the vertex u in G. G is represented by an adjacency matrix a[][].

*a) Viết hàm có tên DFS(int u) bằng C/C++ sử dụng ngăn xếp thực hiện thuật toán tìm kiếm theo chiều sâu bắt đầu từ đỉnh u trên đồ thị G = <V, E> được biểu diễn dưới dạng ma trận kề a[ ][ ].*

b) Utilize the previously written function DFS(int u) to show a path from vertex 1 to vertex 5 in the graph G from Question 1. Show the results of each step of the algorithm.

*b) Sử dụng thuật toán tìm kiếm theo chiều sâu DFS vừa trình bày chỉ ra đường đi từ đỉnh 1 đến đỉnh 5 trên đồ thị G cho trong Câu 1. Chỉ rõ kết quả tại từng bước thực hiện thuật toán.*

**Question 3** (2 marks)

Given an undirected graph G = <V, E> having 10 vertices. G is represented by the adjacency matrix shown below:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 3 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 6 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 9 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 10 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

1. Write the backtracking algorithm, also known as the recursive algorithm, to enumerate all Hamiltonian circuits starting from a vertex u in any graph.
2. Utilize the previously written backtracking algorithm to display the Hamiltonian circuit search tree and list all Hamiltonian circuits starting from vertex 5 of the given graph G.

*Cho đồ thị vô hướng G = <V, E> gồm 10 đỉnh được biểu diễn dưới dạng ma trận kề như sau:*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* |
| *1* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *1* | *1* | *0* |
| *2* | *0* | *0* | *1* | *0* | *0* | *1* | *0* | *0* | *0* | *1* |
| *3* | *0* | *1* | *0* | *1* | *0* | *1* | *0* | *0* | *0* | *0* |
| *4* | *0* | *0* | *1* | *0* | *1* | *0* | *1* | *0* | *0* | *0* |
| *5* | *0* | *0* | *0* | *1* | *0* | *0* | *1* | *0* | *0* | *0* |
| *6* | *0* | *1* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *0* |
| *7* | *0* | *0* | *0* | *1* | *1* | *0* | *0* | *1* | *1* | *0* |
| *8* | *1* | *0* | *0* | *0* | *0* | *0* | *1* | *0* | *1* | *0* |
| *9* | *1* | *0* | *0* | *0* | *0* | *0* | *1* | *1* | *0* | *1* |
| *10* | *0* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *1* | *0* |

1. *Trình bày thuật toán quay lui (đệ quy) tìm các chu trình Hamilton bắt đầu tử 1 đỉnh u của một đồ thị cho trước.*
2. *Áp dụng thuật toán vừa trình bày, xây dựng cây tìm kiếm các chu trình Hamilton và chỉ ra các chu trình Hamilton xuất phát từ đỉnh u = 5 trên đồ thị G đã cho.*

**Question 4** (2 marks)

Given an undirected graph G = <V, E> having 10 vertices. G is represented by the weight matrix shown below:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 2 | ꝏ | ꝏ | 1 | 2 | 7 | ꝏ | 9 | 11 |
| 2 | 2 | 0 | 3 | ꝏ | 1 | ꝏ | ꝏ | ꝏ | ꝏ | 11 |
| 3 | ꝏ | 3 | 0 | 4 | 1 | ꝏ | ꝏ | ꝏ | ꝏ | ꝏ |
| 4 | ꝏ | ꝏ | 4 | 0 | 1 | 4 | ꝏ | ꝏ | ꝏ | ꝏ |
| 5 | 1 | 1 | 1 | 1 | 0 | 1 | ꝏ | ꝏ | ꝏ | ꝏ |
| 6 | 2 | ꝏ | ꝏ | 4 | 1 | 0 | 7 | ꝏ | ꝏ | ꝏ |
| 7 | 7 | ꝏ | ꝏ | ꝏ | ꝏ | 7 | 0 | 7 | 7 | ꝏ |
| 8 | ꝏ | ꝏ | ꝏ | ꝏ | ꝏ | ꝏ | 7 | 0 | 8 | 8 |
| 9 | 9 | ꝏ | ꝏ | ꝏ | ꝏ | ꝏ | 7 | 8 | 0 | 9 |
| 10 | 11 | 11 | ꝏ | ꝏ | ꝏ | ꝏ | ꝏ | 8 | 9 | 0 |

1. Write the Prim algorithm to construct the minimum spanning tree starting from one vertex u in an undirected weighted connected graph.
2. Utilize the previously written Prim algorithm to display the minimum spanning tree and its length. The tree starts from vertex 6 of the given graph G. Show the result of each step of the algorithm.

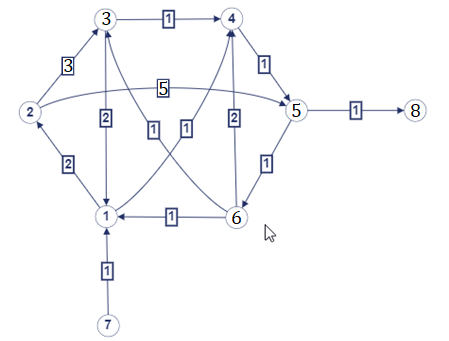
*Cho đồ thị vô hướng G = <V, E> gồm 10 đỉnh được biểu diễn dưới dạng ma trận trọng số như sau:*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* |
| *1* | *0* | *2* | *ꝏ* | *ꝏ* | *1* | *2* | *7* | *ꝏ* | *9* | *11* |
| *2* | *2* | *0* | *3* | *ꝏ* | *1* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* | *11* |
| *3* | *ꝏ* | *3* | *0* | *4* | *1* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* |
| *4* | *ꝏ* | *ꝏ* | *4* | *0* | *1* | *4* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* |
| *5* | *1* | *1* | *1* | *1* | *0* | *1* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* |
| *6* | *2* | *ꝏ* | *ꝏ* | *4* | *1* | *0* | *7* | *ꝏ* | *ꝏ* | *ꝏ* |
| *7* | *7* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* | *7* | *0* | *7* | *7* | *ꝏ* |
| *8* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* | *7* | *0* | *8* | *8* |
| *9* | *9* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* | *7* | *8* | *0* | *9* |
| *10* | *11* | *11* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* | *ꝏ* | *8* | *9* | *0* |

1. *Trình bày thuật toán Prim xuất phát từ 1 đỉnh u cho trước xây dựng cây khung nhỏ nhất của đồ thị vô hướng, liên thông, có trọng số.*
2. *Áp dụng thuật toán Prim chỉ ra độ dài cây và các cạnh của cây khung nhỏ nhất của đồ thị G xuất phát từ đỉnh 6. Chỉ rõ kết quả tại mỗi bước thực hiện thuật toán.*

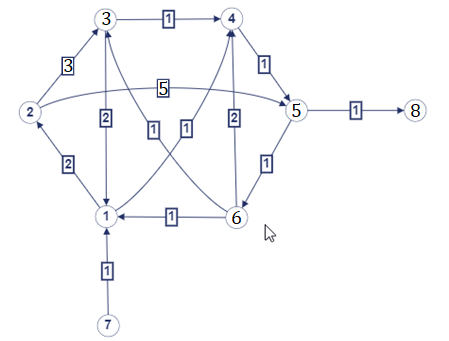
**Question 5** (2 marks)

Given a directed weighted graph G = <V, E> shown below:



1. Write the DIJKSTRA(int u) function in C/C++ to showcase the Dijkstra algorithm for finding the shortest path and its length d[v] from a given vertex u to other vertices v in a graph G = <V, E>. The graph is represented by a weight matrix a[][].
2. Utilize the previously implemented DIJKSTRA(int u) function to determine the shortest path and its length d[v] from a given vertex u = 2 to other vertices v in the given graph G = <V, E> depicted in the figure.

*Cho đồ thị có hướng G = <V, E> như hình dưới, trọng số được ghi trên mỗi cung.*

**

1. *Viết hàm có tên* ***DIJKSTRA****(int u) trên C/C++ mô tả thuật toán Dijkstra tìm khoảng cách d[v] và đường đi ngắn nhất xuất phát từ đỉnh u đến các đỉnh v của đồ thị G = <V, E> được biểu diễn dưới dạng ma trận trọng số a[ ][ ].*
2. *Áp dụng thuật toán Dijkstra chỉ ra khoảng cách và đường đi ngắn nhất từ đỉnh số 2 đến các đỉnh của đồ thị G.*

***No*te:** *This is a closed-book exam. Students are not allowed to use any material or resources.*

*Student name*:……………………………… *Class*:…………………*Test room*:………..

*Student signature:*………………………………………