***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**: 2

**Question 1** (2 marks)

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

Adj(1) = {2, 6, 10} Adj(6) = {1, 4, 5, 7}

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

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

Adj(4) = {3, 5, 6} Adj(9) = {3, 8, 10}

Adj(5) = {4, 6, 7, 8} Adj(10) = {1, 9}

1. Determine the degree of the vertices of G.
2. Represent G using the vertex-edge incidence matrix.

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

*Ke(1) = {2, 6, 10} Ke(6) = {1, 4, 5, 7}*

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

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

*Ke(4) = {3, 5, 6} Ke(9) = {3, 8, 10}*

*Ke(5) = {4, 6, 7, 8} Ke(10) = {1, 9}*

*a) Tìm bậc của mỗi đỉnh trên đồ thị.*

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

**Question 2** (2 marks)

a) Write a function called BFS(int u) in C/C++ based on the queue data structure. The BFS(int u) function does the breadth 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 BFS(int u) bằng C/C++ (sử dụng hàng đợi) thực hiện thuật toán tìm kiếm theo chiều rộng bắt đầu từ đỉnh u của đồ thị G = <V, E> được biểu diễn dưới dạng ma trận kề a[ ][ ].*

b) Utilize the previously written function BFS(int u) to identify all bridges in the graph G from Question 1. Note that listing all intermediate results of the breadth-first search is not necessary. Only the final result of the breadth-first search should be provided.

*b) Sử dụng thuật toán duyệt theo chiều rộng (BFS) vừa trình bày, duyệt toàn bộ cạnh cầu trên đồ thị G cho trong Câu 1? (Không cần ghi chi tiết kết quả của các bước thực hiện thuật toán BFS, chỉ cần ghi kết quả duyệt BFS trong các bước thực hiện thuật toán duyệt tìm cạnh cầu).*

**Question 3** (2 marks)

Given a directed 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 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 9 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 10 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 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 7 of the given graph G.

*Cho đồ thị có 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* | *1* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *0* |
| *2* | *0* | *0* | *1* | *1* | *0* | *0* | *1* | *0* | *0* | *0* |
| *3* | *0* | *0* | *0* | *1* | *1* | *0* | *0* | *0* | *0* | *0* |
| *4* | *0* | *0* | *0* | *0* | *1* | *1* | *0* | *0* | *0* | *0* |
| *5* | *0* | *1* | *0* | *0* | *0* | *1* | *0* | *0* | *0* | *1* |
| *6* | *0* | *0* | *0* | *0* | *0* | *0* | *1* | *1* | *0* | *0* |
| *7* | *0* | *0* | *0* | *0* | *1* | *0* | *0* | *1* | *1* | *0* |
| *8* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *1* | *1* |
| *9* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *1* |
| *10* | *1* | *1* | *0* | *0* | *0* | *0* | *1* | *0* | *0* | *0* |

1. *Trình bày thuật toán quay lui (đệ quy) tìm các chu trình Hamilton xuất phát từ 1 đỉnh u của một đồ thị cho trước.*
2. *Áp dụng thuật toán vừa trình bày, chỉ ra cây tìm kiếm các chu trình Hamilton và ghi ra các chu trình Hamilton xuất phát từ đỉnh 7 trên đồ thị G.*

**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 | 1 | ꝏ | ꝏ | ꝏ | 1 | ꝏ | ꝏ | 1 | 1 |
| 2 | 1 | 0 | 1 | 1 | ꝏ | ꝏ | ꝏ | 1 | ꝏ | 1 |
| 3 | ꝏ | 1 | 0 | 2 | 2 | ꝏ | ꝏ | ꝏ | ꝏ | ꝏ |
| 4 | ꝏ | 1 | 2 | 0 | 3 | 3 | ꝏ | ꝏ | ꝏ | ꝏ |
| 5 | ꝏ | ꝏ | 2 | 3 | 0 | 4 | 4 | ꝏ | ꝏ | ꝏ |
| 6 | 1 | ꝏ | ꝏ | 3 | 4 | 0 | 5 | ꝏ | ꝏ | ꝏ |
| 7 | ꝏ | ꝏ | ꝏ | ꝏ | 4 | 5 | 0 | 6 | 6 | ꝏ |
| 8 | ꝏ | 1 | ꝏ | ꝏ | ꝏ | ꝏ | 6 | 0 | 7 | 7 |
| 9 | 1 | ꝏ | ꝏ | ꝏ | ꝏ | ꝏ | 6 | 7 | 0 | ꝏ |
| 10 | 1 | 1 | ꝏ | ꝏ | ꝏ | ꝏ | ꝏ | 7 | ꝏ | 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 8 of the given graph G. Show the result of each step of the algorithm.

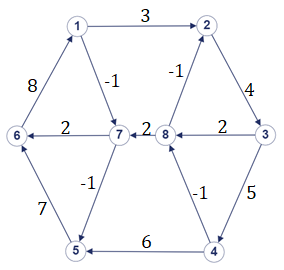
*Cho đồ thị vô hướng G = <V, E> gồm 10 đỉnh được biểu diễn dưới dạng ma trận trọng số như sau:*

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

1. *Trình bày thuật toán Prim tìm cây khung nhỏ nhất xuất phát từ 1 đỉnh u của đồ thị vô hướng, liên thông, có trọng số.*
2. *Áp dụng thuật toán Prim vừa trình bày, chỉ ra độ dài cây và các cạnh của cây khung nhỏ nhất của đồ thị G xuất phát từ đỉnh 8, chỉ rõ kết quả tại mỗi bước thực hiện thuật toán.*

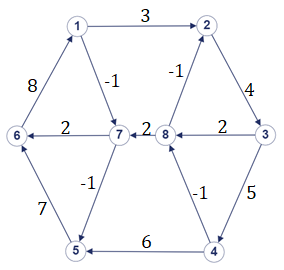
**Question 5** (2 marks)

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



1. Write the BELLMAN(int u) function in C/C++ to showcase the Bellman-Ford 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 BELLMAN(int u) function to determine the shortest path and its length d[v] from a given vertex u = 5 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 bên cạnh mỗi cung.*

**

1. *Viết hàm có tên là* ***BELLMAN****(int u) bằng C/C++ mô tả thuật toán Bellman-Ford tìm khoảng cách d[v] và đường đi ngắn nhất từ đỉnh u cho trước 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 Bellman-Ford chỉ ra khoảng cách và đường đi ngắn nhất từ đỉnh u = 5 đến các đỉnh của đồ thị G đã cho trong hình.*

***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:*………………………………………