Cairo University

Faculty of Computers and Artificial Intelligence

SCS316-Algorithms Analysis and Design
Assignment - 3

Supervised By:

DR. Zeinab Abd ElHaliem

TA. Dalia Maher Mohamed

Prepared & Implemented By:

Mootaz Medhat Ezzat Abdelwahab (20206074)

Academic Year: 2023/2024

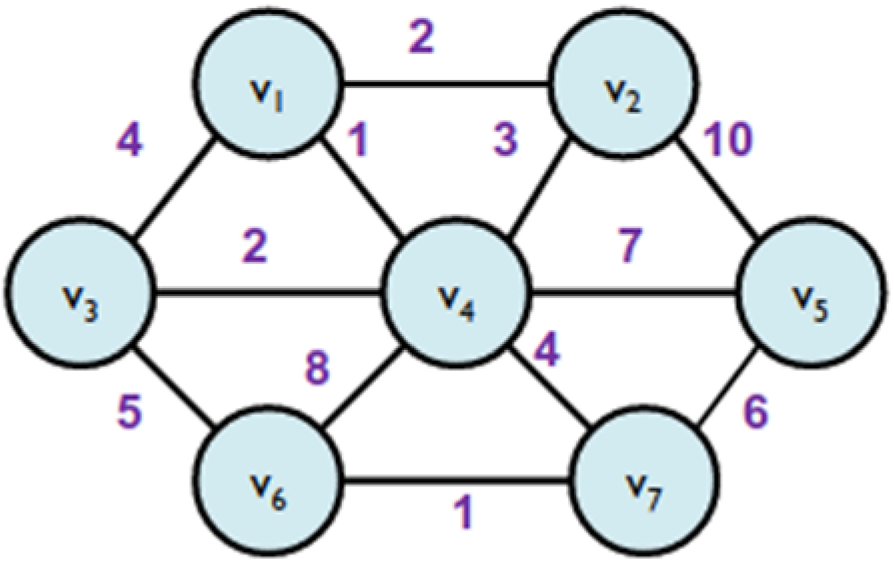
Second Semester

SCS316-Algorithms Analysis and Design [Assignment - 3]

➤ Question (1): [MST]

Please write an MST Prim's code to solve the following problem with the same input and print the output graph including nodes and vertices.

Given Graph G:



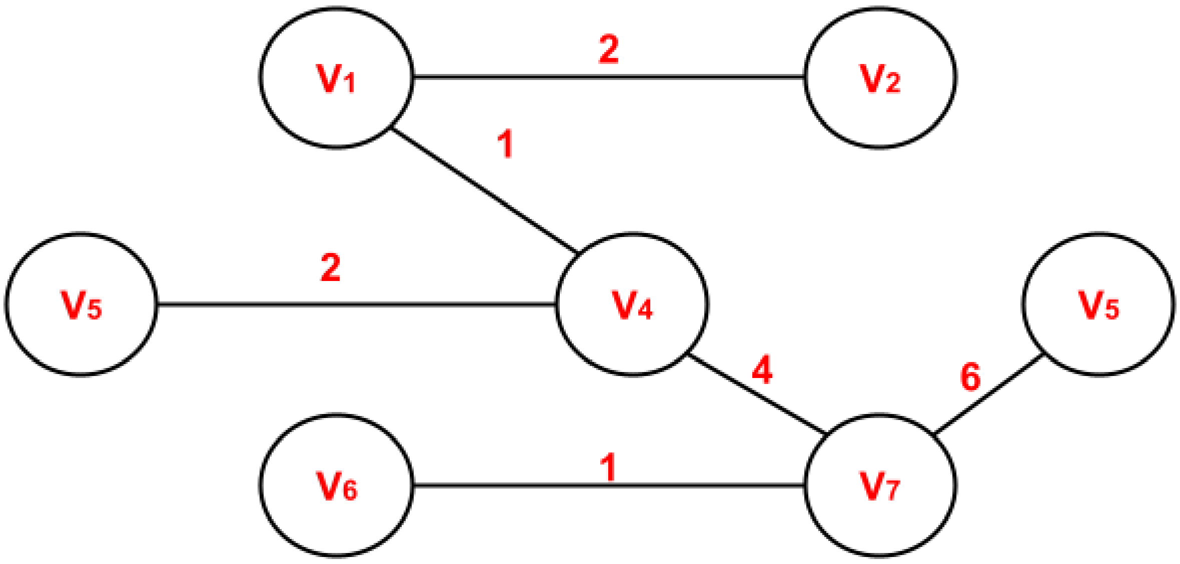
Determine the MST, using Prim's starting with vertexV1:

■ **The idea of Prim's Minimum Spanning Tree (MST) Algorithm :-**

- **Prim's MST algorithm is a Greedy algorithm.**
- **It finds a minimum spanning tree for a weighted undirected graph.**
- **In Prim's MST algorithm we create a MST by picking edges that have minimum weight one by one. To do this we maintain two sets (two arrays of vertices) :-**
  - **a set (array) of the vertices already included in MST.**
  - **a set (array) of the vertices not yet included in MST.**
- **The Greedy algorithm (choice) here is to pick the edge with minimum weight that connects the two sets (two arrays of vertices) one by one at a time.**

■ **Solution :-**

• **In this problem we used adjacency matrix to represent the graph, Then the time complexity will be O(|V|*|V|) and It can be reduced to O(|E|*(log|V|)) using adjacency list and binary heap.**



Total Weight (Cost) = 1 + 2 + 2 + 4 + 1 + 6 = 16

➢ Question (2): [Greedy]

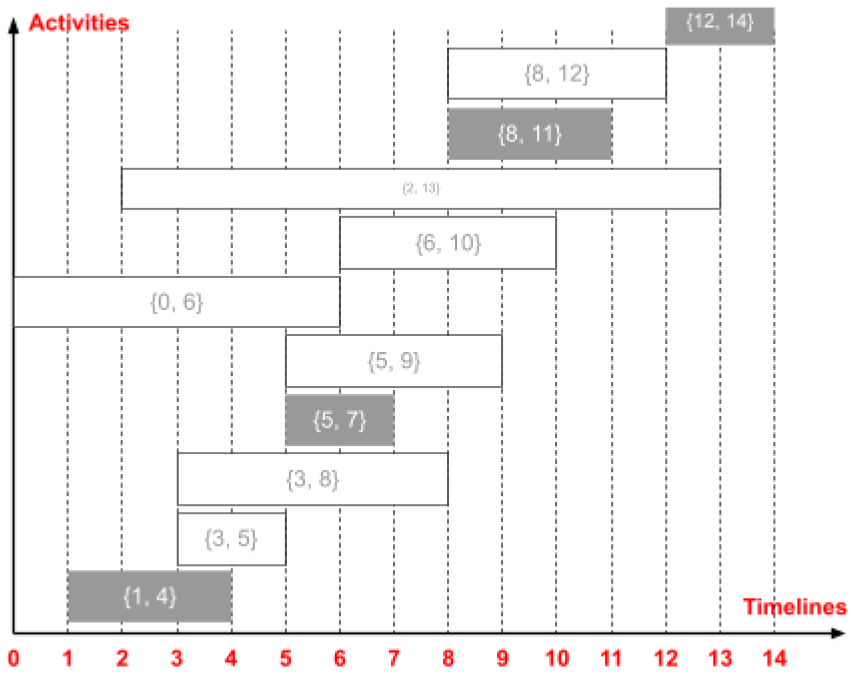
Given a set of activities and the starting and finishing time of each activity, find the maximum number of activities that can be performed by a single person assuming that a person can only work on a single activity at a time. This problem is called the activity selection problem, which concerns the selection of non-conflicting activities to perform within a given time frame, given a set of activities each marked by a start and finish time.

Input:
11
{1, 4}, {3, 5}, {0, 6}, {5, 7}, {3, 8}, {5, 9}, {6, 10}, {8, 11}, {8, 12}, {2, 13}, {12, 14}

Output:
4
{1, 4}, {5, 7}, {8, 11}, {12, 14}

▪ Solution :-

- **In this problem we assumed that the activities will be sorted according to their finish time, So the first activity always be selected.**
- **The Greedy algorithm (choice) here is to always pick the next activity that has the smallest finish time and it's start time is more than or equal to finish time of the previously selected activity.**
- **Time complexity for this problem will be O(N).**
- **N refers to the total number of activities.**



▪ Note :-

If the given set of activities is unsorted then we will have to use the sort( ) method so the time complexity will vary depending on the time complexity of the used sort method.

➢ Question (3): [Greedy]

Given an array of size n, each element contains either a 'P' for a policeman or a 'T' for a thief. Find the maximum number of thieves that the police can catch.

Keep in mind the following conditions:
1. Each policeman can catch only one thief.
2. A policeman cannot catch a thief who is more than K units away from him.

■ Example 1:
Input: N = 5, K = 1
arr[] = {P, T, T, P, T}
Output: 2

Explanation: Maximum 2 thieves can be caught. The first policeman catches the first thief and the second policeman can catch either the second or third thief.

■ Example 2:
Input: N = 6, K = 2
arr[] = {T, T, P, P, T, P}
Output: 3

Explanation: Maximum of 3 thieves can be caught.

■ Solution :-

- An explanation of the algorithm for this problem is mentioned in detail in the comments in the file p3.cpp file.

- Time complexity for this problem will be O(N).

- N refers to the total number of policemen and thieves.

➢ Question (4): [Dynamic Programming]

Given a string s, return the longest palindromic substring in s using dynamic programming.

- **Note: A string is called a palindrome string if the reverse of that string is the same as the original string. For example radar and level.**

- **Example 1:**
  Input : s = “cbbd”
  Output: bb

➥ **Solution :-**

- **By applying Dynamic Programming in this problem the time complexity reduced from O(N^3) to O(N^2) by storing results of subProblems and it cab be reduced to O(N) if the reverse of that string is the same as the original string.**
- **N refers to the length of the given string.**

str = cbbd

		0	1	2	3
		c	b	b	d
0	c	1	0	0	0
1	b	0	1	1	0
2	b	0	0	1	0
3	d	0	0	0	1