Algorithms Analysis & Design

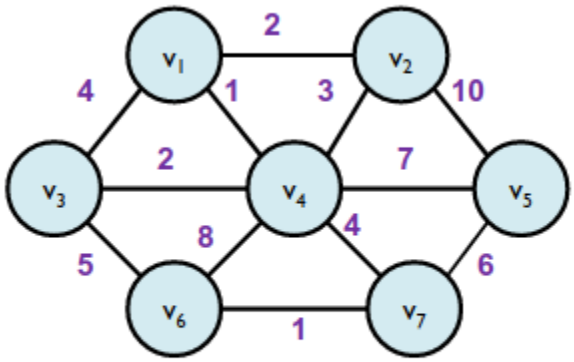
Assignment #3

Guidelines:

- You should submit the assignment in a group of a maximum of 4 students.
- You must join the same team of assignment 2.
- The discussions of both assignments 2 and 3 will be held together.
- submit a zip file (name this file with your ids: A1_ID1_ID2_ID3_ID4.zip) with all files (codes) for each problem (name code file p1, p2, .. etc)

Question (1): [MST] (30 grades)

Given Graph G:



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

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.

Question (2) [Greedy] (20 grades)

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}

Question (3) [Greedy] (20 grades)

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.

Question (4) [Dynamic Programming] (30 grades)

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