# National University of Computer and Emerging Sciences, Lahore Campus

| | | | |
|---|---|---|---|
| Course: | Design and Analysis of Algorithms | Course Code: | CS 302 |
| Program: | BS (Computer Science) | Semester: | Fall 2017 |
| Duration: | 60 Minutes | Total Marks: | 40 |
| Paper Date: | 04-Nov-2017 | Page(s): | 2 |
| Section: | N/A | Section: | |
| Exam: | Midterm Exam 1 | Roll No: | |

**Instruction/Notes:** Solve this exam on the answer sheet.

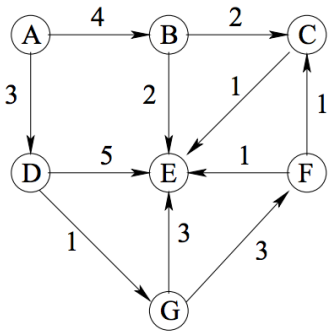
**Question 1 (15 points)**

A robot, initially standing at point 0, needs to travel from point n on a straight line. In order to do that it can take steps of either size 1, size 2 or size 3. For example, if it is at point k, then it can go next to point k+1, point k+2 or point k+3. This means that there are many different ways in which it can get from point 0 to point n. For example, if n=3, there are 4 ways to get from 0 to 3: either take three steps of 1, or take a step of 2 followed by step of 1, or take a step of 1 followed by a step of 2, or take a single step of 3.
We wish to write a dynamic programming algorithm to compute the total number of ways in which the robot can get from point 0 to point n.

(a) How many ways are there for the robot to get from 0 to 4?
(b) Let V[k] be defined as the number of ways for the robot to get from point 0 to point k. Write a recurrence for V[k] in terms of smaller sub-problems. Also give the base case(s).
(c) Make the array V from v[0] to v[8] and fill in the values using the recurrence in (b).

**Question 2 (15 points)**

Perform Dijkstra's shortest path algorithm on the following directed graph, taking A as source. List the vertices in the order in which they are deleted from the min-heap and their shortest path lengths from A and their parent node according to the algorithm. For example, the first of be deleted is [A, 0, -], i.e. vertex A with path length 0 and no parent. There is no need to show the contents of the entire heap or any other information.



**P.T.O.**

**Question 3 (10 points)**

Given an undirected graph $G = (V, E)$, and an edge $e \in E$ write an algorithm which determines whether $G$ contains a cycle containing the edge $e$. The algorithm should take no more than $O(|V| + |E|)$ time. Describe your algorithm in English in a few lines, then give pseudo code. *There is no partial credit in this problem.*

THE END