
Instructions

1. This is an individual assignment.
 2. Your code must completely be your own. You are not allowed to take guidance from any general-purpose code or problem specific code meant to solve these or related problems. Remember, it is easy to detect this kind of plagiarism.
 3. All the PROBLEMS are COMPULSORY.
 4. **Write only a single main function.** You can call the required functions from the main function.
 4. Name the file as follows: S2021xxxxx_Ao8.c
 5. DO NOT zip. **Upload a single .c file** directly to your submission in the common Google classroom.
-

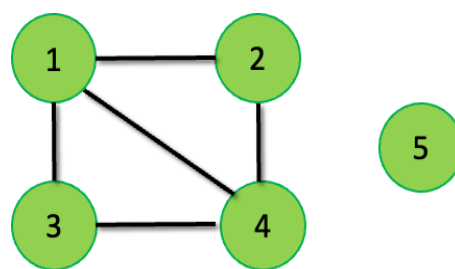
Question 1: [4 points]

Create a simple hash table using an array. Start with the array size of 5 and whenever the hash table reaches 80% capacity double the size of the array each time. Insert the values (integers) into the hash table using modular hashing [mod by the table size]. Print the values in the hash table after each insertion. For collision resolution, use open addressing with linear probing.

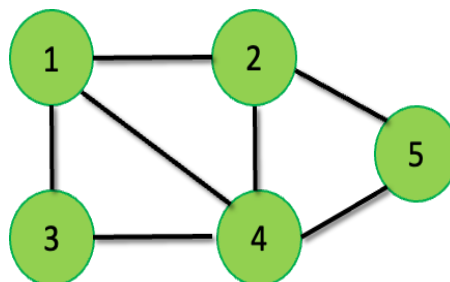
Question 2: [3 points]

Implement an algorithm that can generate the corresponding adjacency matrix for the given graphs. The algorithm should also find out whether the given undirected graphs are connected or not.

G1:



G2:



Question 3: [3 points]

Implement an algorithm that can generate the corresponding adjacency list for the given graphs. Each adjacency list should be a linked list storing the numbers of the neighboring vertices. For an undirected edge (u, v) , store both u in the adjacency list of v , and v in the adjacency list of u . Print the adjacency lists for each vertex.