CSC 226 SUMMER 2020
ALGORITHMS AND DATA STRUCTURES II
ASSIGNMENT 2
UNIVERSITY OF VICTORIA

1. Draw the 11-item hash table resulting from hashing the keys 12, 44, 13, 88, 23 94, 11, 39, 20, 16, and 5, using hash function $h(k) = (2k + 5) \bmod 11$, assuming collisions are handled by each of the following:
   a. Separate chaining.
   b. Linear probing.
   c. Quadratic probing up to the point where the method fails because no empty slot is found.
   d. Double hashing using the secondary hash function $h'(k) = 7 - (k \bmod 7)$.

2. Give a pseudocode description for performing insertion, searching, removal from a hash table that uses quadratic probing to resolve collisions where we use a special marker to represent deleted elements. You may use the pseudocode given on page 201 of the Goodrich and Tamassia book (attached to this assignment) as a template for your pseudocode. Note, you will not use the Shift(i) method.

3. Write a version of the depth-first search algorithm, where graph G is represented by an adjacency matrix and the output the preorder labelling of the vertices as an integer array. You can use the code on page 367 of Goodrich and Tamassia as a starting point (given in the slides). Assume the graph is simple and connected. Do not worry about labelling the edges as explored, discovery or back edges. Focus on how you would use the adjacency matrix to find adjacent vertices and how to determine if a vertex has been explored or not yet (Hint: the preorder labels may help with that.)

   Algorithm matrixDFS(G,v)
   Input: A graph G, with n vertices labeled $0, \dots, n-1$, represented as an adjacency matrix and a starting vertex v.
   Output: An integer array of size n, containing the preorder labelling of the vertices.

4. Analyze the best-case and worst-case running times of your matrixDFS() algorithm in question 3 above. How does this compare to the best-case and worst-case running times of implementing DFS with an adjacency list representation of the graph?

5. Consider the following directed graph, G, with vertices labelled 1 to 6.

$$G = (\{1,2,3,4,5,6\}, \{(1,2), (1,5), (3,1), (3,4), (4,5), (5,6), (6,3)\})$$

   a. Draw graph G and give its adjacency matrix representation, labeling it G_0.
   b. Run the Floyd-Warshall transitive closure algorithm on G, resulting in the new graph G^*. Show the contents of the adjacency matrix after each iteration of the outside loop of the algorithm (that is, show G_k for each $k = 1, \dots, 6$).