# Lab08-Graph Exploration

CS214-Algorithm and Complexity, Xiaofeng Gao & Lei Wang, Spring 2021.

$*$ If there is any problem, please contact TA Yihao Xie.
$*$ Name:_________ Student ID:_________ Email: ____________

1. Given a graph $G = (V, E)$. Prove the following propositions.

   (a) Let $e$ be a maximum-weight edge on some cycle of connected graph $G = (V, E)$. Then there is a minimum spanning tree of $G$ that does not include $e$. Moreover, there is no minimum spanning tree of $G$ that includes $e$ if $e$ is the unique maximum-weight edge on the cycle.

   (b) Let $T$ and $T'$ are two different minimum spanning trees of $G$. Then $T'$ can be obtained by recursively substitute one edge in $T$ by one edge in $T'$.

2. Let $G = (V, E)$ be a connected, undirected graph. Give an $O(v + E)$-time algorithm to compute a path in $G$ that traverses each edge in $E$ exactly once in each direction. Describe how you can find your way out of a maze if you are given a large supply of pennies.

3. Consider the maze shown in Figure 1. The black blocks in the figure are blocks that can not be passed through. Suppose the block are explored in the order of right, down, left and up. That is, to go to the next block from $(X, Y)$, we always explore $(X, Y + 1)$ first, and then $(X + 1, Y)$,$(X, Y - 1)$ and$(X - 1, Y)$ at last. Answer the following subquestions:

   (a) Give the sequence of the blocks explored by using DFS to find a path from the "start" to the "finish".

   (b) Give the sequence of the blocks explored by using BFS to find the shortest path from the "start" to the "finish".

   (c) Consider a maze with a larger size. Discuss which of BFS and DFS will be used to find one path and which will be used to find the shortest path from the start block to the finish block.

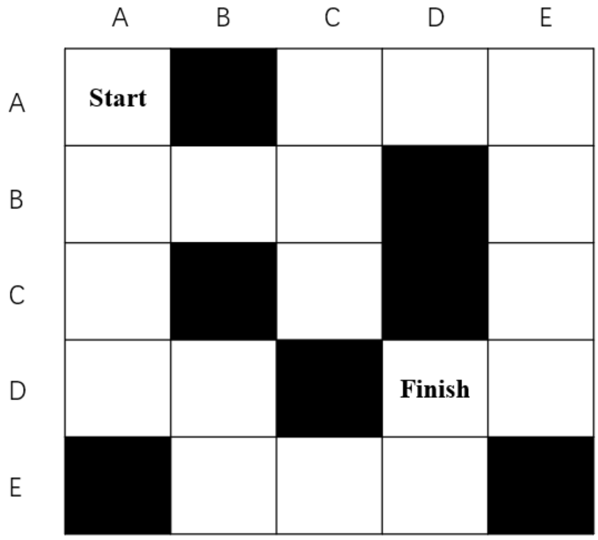

Figure 1: An example of making room for one new element in the set of arrays.

4. Given a directed graph $G$, whose vertices and edges information are introduced in data file "SCC.in". Please find its number of Strongly Connected Components with respect to the following subquestions.

   (a) Read the code and explanations of the provided C/C++ source code "SCC.cpp", and try to complete this implementation.

   (b) Visualize the above selected Strongly Connected Components for this graph $G$. Use the *Gephi* or other software you preferred to draw the graph. (If you feel that the data provided in "SCC.in" is not beautiful, you can also generate your own data with more vertices and edges than $G$ and draw an additional graph. Notice that results of your visualization will be taken into the consideration of Best Lab.)

**Remark:** Please include your .pdf, .tex, .cpp files for uploading with standard file names.