

1. Implement Graph Class for unweighted, undirected Graph with Adjacency Matrix with following functions
 - a. Add Edge
 - b. Delete Edge
 - c. BFS
 - d. DFS
 - e. `vector<int> getShortestPath(int src, int dest)`
 - f. `vector<vector<int> > findAllConnectedComponents()`
 - g. `bool isConnected()`
 - h. `bool isCyclic()`
 - i. `bool isBipartite()` – read from the internet
2. Given a snake and ladder board, find the minimum number of dice throws required to reach the destination or last cell from source or 1st cell. Basically, the player has total control over outcome of dice throw and wants to find out minimum number of throws required to reach last cell. If the player reaches a cell which is base of a ladder, the player has to climb up that ladder and if reaches a cell is mouth of the snake, has to go down to the tail of snake without a dice throw
3. Given a number of friends who have to give or take some amount of money from one another. Design an algorithm by which the total cash flow among all the friends is minimized
4. <http://www.cse.iitd.ernet.in/~naveen/courses/CSL201/assgn6.html>
5. Write an algorithm that receives a dictionary (Array of words). The array is sorted lexicographically, but the abc order can be anything. For example, it could be z, y, x, ..., c, b, a. Or it could be completely messed up: d, g, w, y, ... It doesn't even need to include all the abc letters, and finally it doesn't have to be letters at all. It could be any symbols that form a string. For example it could be made up of 5, α, !, @, θ... You get the idea. It's up to your algorithm to discover what the letters are (easy part). The algorithm should return the correct lexicographic order of the symbols.
6. You're a photographer for a soccer meet. You will be taking pictures of pairs of opposing teams. All teams have the same number of players. A team photo consists of a front row of players and a back row of players. A player in the back row must be taller than the player in front of him. All players in a single row are on the same team. First, design an algorithm that takes as input two teams and the heights of the players in the teams and checks if it is possible to place players to take the photo subject to

the placement constraint. Then, generalize your solution: if you are given n teams, how can you determine the largest number of teams that can be photographed simultaneously subject to the same constraints?

7. Given a boolean 2D matrix, find the number of islands. A group of connected 1s forms an island. For example, the below matrix contains 5 islands

```
{1, 1, 0, 0, 0},  
{0, 1, 0, 0, 1},  
{1, 0, 0, 1, 1},  
{0, 0, 0, 0, 0},  
{1, 0, 1, 0, 1}
```