Birla Institute of Technology and Science, Pilani

Second Semester 2017-2018, DSA (CS F211)

Lab Assignment #8

1. Suppose you are given a set of points (x and y coordinates), find the convex polygon with minimum number of edges and minimum area that includes all the points within it. No three points are collinear.
   Input:
   N – points
   N lines of x and y coordinates.
   Output:
   M – number of edges required.
   M lines of x and y coordinates between which the edges lie.
   Time Complexity: O(NlogN)
   Sample Input:
   6
   0 0
   0 4
   -4 0
   5 0
   0 -6
   1 0
   Sample Output:
   4
   -4 0
   5 0
   0 -6
   0 4
   Explanation: If we draw the four line segments given, they include all the 6 points and any other polygon would be of greater area than this.

2. The travelling salesman is tired of travelling. Now he wants to figure out the number of ways of going from place to place. He has a map of roads connecting different cities. Help him out in doing this: Given two cities, output the number of different paths between them that take a maximum of k roads.
   Input:
   M – number of cities K – maximum number of roads to take
   M * M matrix which has 0s and 1s. 0 at M_{ij} means that there is no direct road from city I to city j. 1 at M_{ij} means otherwise.
   i j – cities numbers between which the number of paths has to be calculated.
   Output:
   X – number of paths with maximum k roads from city i to city j.
   Time Complexity: O(M^3logK)
   Sample Input:

4 2
0 1 1 1
0 0 0 1
0 0 0 1
0 0 0 0
0 3

Sample Output:

3

Explanation:

City 0 to City 3 can be reached in 3 ways all of which take 2 or less roads:

a. Directly from 0 to 3
b. Through 1
c. Through 2

3. Dharm is a very rich guy and he is fond of fancy cars. He recently bought a few and wants to test them out. He needs a very long stretch of road to do so. He has the map which shows distances from one point to another. He has to determine the longest ride he can have. Help him out. It is ensured that there are no cyclic paths.

Input:

M – number of locations which have roads between them.
M*M matrix in which M_{ij} contains the length of road between location i and j.

Output:

X – length of the longest path.

Time Complexity:

$O(M^3 \log M)$

Sample Input:

4
N 1 8 N
N N 5 7
N N N 2
N N N N
(N – no path)

Sample Output:

10

Explanation:

Path from 1 to 4 passing through 3 is the longest. 10 = 8 + 2.

4. Mr. Dubey is writing a software. His program starts from an initial state (say state 0) and goes through different states before finishing. The program searches for the least-cost path to reach the final state (which can be any of the states other than the initial one). Given the costs (costs can be negative) to transition between states, determine if Dubey's program will ever go into an infinite loop or not.

Input:

M – number of states
M * M matrix – shows the transition cost

Output:

0 – if no possibility of program going into a loop.
1 – if there is a possibility of program going into an infinite loop.

Time Complexity: $O(M^3)$
Sample Input:
5
N 1 N N N
N N 3 N N
N N N -2 N
N -2 N N 10
N N N N N
(N – no path)
Sample Output:
1
Explanation:
There is a loop through 1-2-3-1 states which has a total cost which is negative. So, the program goes through this loop infinitely and the cost keeps on reducing.

5. Himanshu is playing in Takeshi's castle and he is in a Honeycomb Maze. This honeycomb has 4 sided cells only. Each cell is either empty, a wall, a cell with a monster, an appear switch or a disappear switch. Entering the cells with disappear switches makes the monsters disappear and entering the ones with appear switches makes the monsters appear in the same old places. He can enter empty cells and switch cells. He can also enter the monster cells, provided they have disappeared. He is initially placed in any one of the empty cells. The target is to reach any edge cell to escape out. He can move to any adjacent cells (not diagonal) cells in one step. You have the initial map of the maze. Determine the minimum number of steps required to get out of the maze. Also determine if it is possible to get out.
Input:
N M – N rows and M columns
N*M matrix containing W-wall, E-empty, H-initial position of Himanshu (empty cell), A-appear switch, D-disappear switch, M-monster cell.
Output:
K – minimum number of steps for him to get out. -1 if not possible.
Sample Input:

5 8
WWWWWWWE
WHMAEEEM
WEWWEWAW
WEDWEEDW
EWWWWWWW

Sample Output:

21

6. Given a graph, check if the graph is a binary tree. If it is a binary tree print "Yes, it is a binary tree". In case it is a binary tree, check if it is a binary Search tree. If yes, print "Yes, it is a binary search tree" on a new line.
Input: You have to hard code the graph and have to consider three cases.
Case 1: Graph with cycle
Case 2: Binary tree but not Binary search tree

Case 3: Binary Search tree

7. A paradox is a statement or group of statements that leads to a contradiction. Consider the following two statements.

   "The statement below is false."
   "The statement above is true."

   If we assume that 1st statement is true, then according to 1st statement the 2nd statement is false. Since the 2nd statement is false and it says that 1st statement is true, 1st statement must be false which is a contradiction. If we assume that 1st statement is false, then the 2nd statement must be true (since 1st statement says 2nd statement is false). Then according to 2nd statement 1st statement must be true which is a contradiction. Thus the statement cannot be classified as true or false. This is a paradox.

   You will be given N statements which are of the form "Statement X is true/false", 1<=X<=N, in a format which will be mentioned below, your task is to find if these set of statements are paradoxical or not. A set of statements is paradoxical if one or more statements cannot be classified as true or false.

   Input:

   Multiple test cases. Each test cases begins with an integer N representing the number of statements. 1<=N<=100. Then follow N lines, representing N statements in the order 1,2,3, ..., N. Each line contains an integer X and a string S separated by a space. 1<=X<=N. S is either "true" or "false" (quotes for clarity), which means the statement is "Statement X is S". N=0 indicates the end of test cases and should not be processed. There are at most 10000 test cases.

   Output:

   For each test case print "PARA" if the set of statements are paradoxical, else print "NOT PARA", in a separate line.

   Sample Input:

   2

   2 false

   1 true

   2

   2 false

   1 false

   0

   Sample Output:

   PARA

   NOT PARA

8. Write a menu-driven C program to implement a hash table of size 10. Consider input keys with hash function h(k) = k mod10. Upon collision, resolve using another hash function h'(k) = 7 – k mod7. And on further, collision use chaining. Your program should support the following operations:
   a. Insert
   b. Print the whole table
   c. Search for an element
   d. Delete an element
   e. Exit
9. Determine the number of connected components in a graph without using recursion.
   Input: N (No of nodes) and M (No of edges)
   M lines follow each line has two Integers X and Y (1 <= X,Y <= N) which denotes that there is an edge between node X and node Y
   Output: No of Connected Components
   Sample Input:
   6 3
   1 2
   2 3
   4 5
   Sample Output:
   3
   Explanation:
   Connected Components are {1,2,3}, {4,5}, {6}
   Time Limit:
   2 Seconds

10. You will be given two prime numbers X and Y. You have to convert X to Y such that in each step you can replace a digit of X with another digit such that the resulting number is also prime. You have to perform this conversion from X to Y in minimum number of steps. Both numbers are four-digit primes (without leading zeros).
    Input:
    T (Number of Test Cases <= 100)
    T lines follow and each test case has two integers X and Y
    Output:
    For each test case output the minimum number of steps to convert X to Y
    Sample Input:

    3
    1033 8179
    1373 8017
    1033 1033
    Sample Output:
    6
    7
    0
    Explanation:
    For 1st test case:

    1033

1733
3733
3739
3779
8779
8179