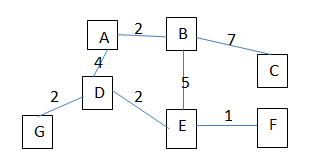
1. Which is a faster graph representation to perform the following operations:

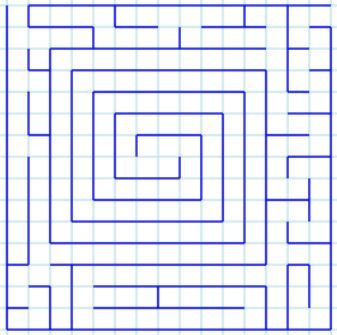
|  |  |
| --- | --- |
| Comparison | Winner |
| test if x is connected to y? | Adjancey matrix |
| find vertex degree? | Adjancey list |
| Less memory on small graphs? | Adjancey list |
| Edge insertion or deletion? | Adjancy matrix |
| Faster to traverse the graph? | Adjancy list |
| Better for most problems? | Adjancy list |

1. A degree of a vertex is the number of other vertices connected to it. Show that the sum of all of the graph vertices degrees is always even if the graph is undirected.

solution:each edge added increase degree by 1 for 2 nodes

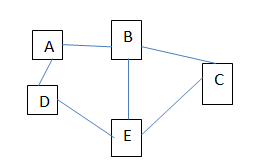
1. Apply both depth first and breadth first algorithms starting from vertex D





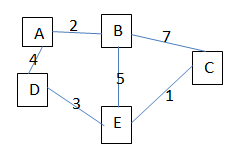
1. For the maze above, Write an algorithm that check if there’s a path to a cell from a given cell.
2. Create an algorithm to determine whether an undirected graph is connected or not (i.e. any node/vertex can be reached from any other node/vertex).
3. For the maze above, Write an algorithm to check how many unique paths are there in the maze.
4. Compute the flight segments sequence and the minimum number of flight segments to get from one city to another one, consider that vertices represent cities, and there is an edge between two vertices whenever there is a flight between the corresponding two cities.
5. Repeat Q7 but try to minimize not the overall number of flights but the total cost of the flights, considered that the weighted edge corresponds to the flight cost.
6. A Computer Science curriculum specifies the prerequisites for each course as a list of courses that should be taken before taking this course. You would like to perform a consistency check of the curriculum, that is, to check that there are no cyclic dependencies. For this, you construct the following directed graph: vertices correspond to courses, there is a directed edge (𝑢, 𝑣) is the course 𝑢 should be taken before the course 𝑣. Then, it is enough to check whether the resulting graph contains a cycle.
7. A spanning tree (ST) is a tree formed from a graph by
   1. Choosing one vertex to be the root
   2. Adding edges connecting the root vertex to the neighbor vertices, so they become children
   3. Doing the same step from the children till all nodes are connected to the tree

Create all possible spanning trees from the shown graph



solution:try playing and every time remove 2 edges.note:for spaning tree with n vertices we have n-1 edges ,focus on cycle and iterate on its edges. **note:if you have complete graph:all vertices are connected together The possible number of spanning trees we can have:v power(v-2 ):** note:spanning tree edges=vertices-1

1. A Minimum Spanning Tree (MST) is a Spanning tree with the minimum sum of edges weights; Find an MST starting from vertex A or Vertex E (is the sum of weights different?)

answer:A-B/A-D/D-E/E-C

1. Apply Dijkstra algorithm in the following problem to find shortest path from A to F and from C to G.

