

EECS 281, Week 6: Wednesday

AVL Trees

Height of a node: _____.

In-order predecessor: _____.

In-order successor: _____.

Left rotation:

Right rotation:

Insert 4, 8, 15, 23, 16 and 42 into an AVL tree.

Delete 8 from the AVL tree.

EECS 281, Week 6: Wednesday

Graphs

Graph: $G = (V, E)$

Set of _____, a.k.a. _____.

Set of _____: pairs of vertices.

Vertices with an edge between are _____.

Vertices or edges may have _____.

A _____ is a sequence of vertices connected by edges.

A _____ is a path whose first and last vertices are the same.

A graph with a cycle is _____. A graph without a cycle is _____.

Two vertices are _____ if there is a path between them.

If all vertices are connected, we say the graph is _____.

Graphs Representations

Adjacency matrix

Adjacency list

Graphs Traversals

Pre-order depth-first traversal

Post-order depth-first traversal

Breadth-first traversal

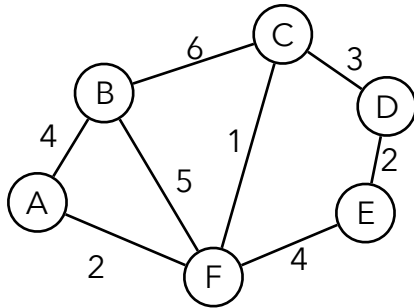
Graph Problems

Shortest path: _____.

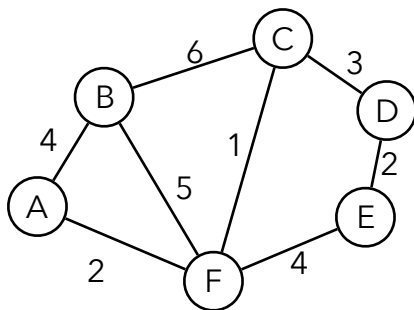
Minimum spanning tree: _____.

Minimum Spanning Tree

Prim's Algorithm



Kruskal's Algorithm



EECS 281, Week 6: Wednesday

Detecting Cycles

Given a *directed* graph represented with an adjacency matrix, determine if the graph has a cycle. Feel free to add helper functions or use an extra page if needed.

```
bool isCyclic(const vector<vector<bool>> &graph) {  
  
    vector<bool> isVisited(graph.size(), false);  
    vector<bool> isOnCurrentPath(graph.size(), false);  
  
    for (size_t i = 0; i < graph.size(); i += 1) {  
        if (dfsVisit(i, graph, isVisited, isOnCurrentPath)) {  
            return true;  
        }  
    }  
  
    return false;  
}  
  
bool dfsVisit(size_t vertexIndex, const vector<vector<bool>> &graph,  
              vector<bool> &isVisited, vector<bool> &isOnCurrentPath) {  
  
    if (!isVisited[vertexIndex]) {  
        isVisited[vertexIndex] = true;  
        isOnCurrentPath[vertexIndex] = true;  
  
        for (size_t adjacentIndex = 0; adjacentIndex < graph.size();  
              adjacentIndex += 1) {  
            if (graph[vertexIndex][adjacentIndex]) {  
                if (!isVisited[adjacentIndex] &&  
                    dfsVisit(adjacentIndex, graph, isVisited, isOnCurrentPath)) {  
                    return true;  
                } else if (isOnCurrentPath[adjacentIndex]) {  
                    return true;  
                }  
            }  
        }  
    }  
    isOnCurrentPath[vertexIndex] = false;  
    return false;  
}
```

What is the worst case time complexity of `isCyclic` in terms of $|V|$ and $|E|$? Explain.

When finished, take a picture of your answers and email it to Maxim.