Week 2: Exercises

Lesson 1

[R] Exercise 1

Implement pre-order, in-order and post-order for arbitrary (non-binary) trees.

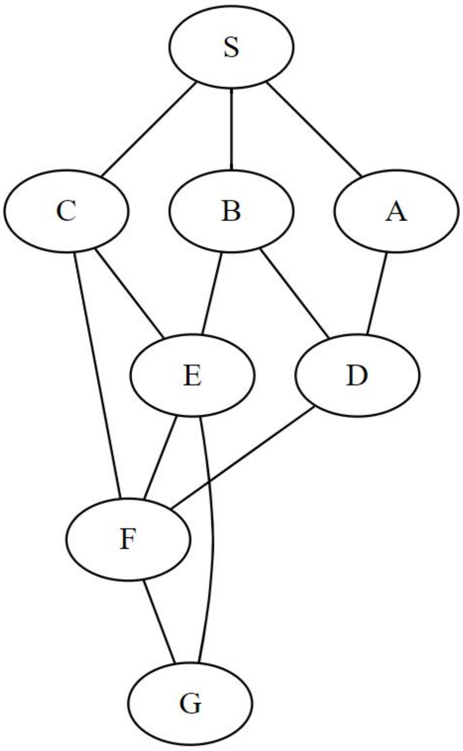
[R] Exercise 2

1. Revisit the “farmer, fox, goose and grain” problem.
   a. Perform depth-first search
   b. Perform breadth-first search
2. Indicate the performed path in the state tree for both algorithms.
3. [E] Perform iterative deepening
4. [F] Implement it in Python.

[R] Exercise 3

Use the graph below

1. Implement it in Python using either adjacency matrix or adjacency list representation
2. Implement iterative deepening to get from node S (start) to node G (goal)

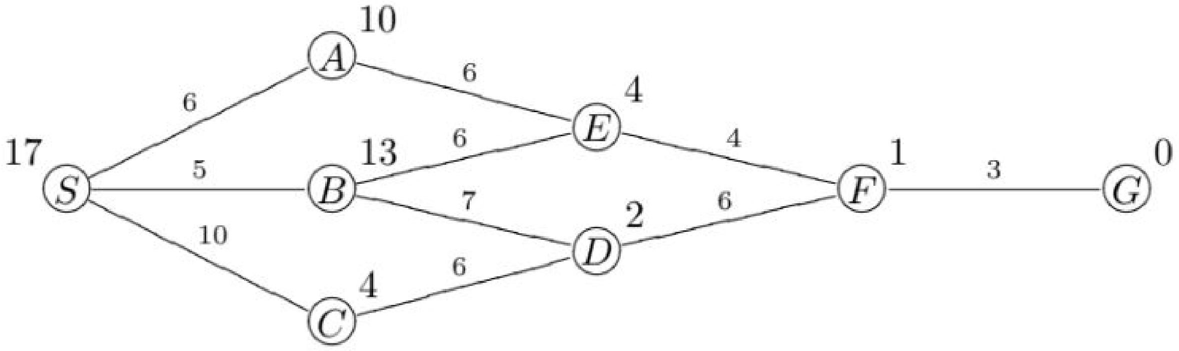


Lesson 2

[M] Exercise 1

Use the graph below.
Implement it in Python using either adjacency matrix or adjacency list representation.

- Implement UCS, (Uniform Cost Search) to get from node S (start) to node G (goal)
- Implement A* to get from node S (start) to node G (goal)



```
graph G {
    S -- A;
    S -- B;
    S -- C;
    A -- E;
    B -- E;
    B -- D;
    C -- D;
    C -- F;
    E -- F;
    D -- F;
    D -- G;
    F -- G;
}
```