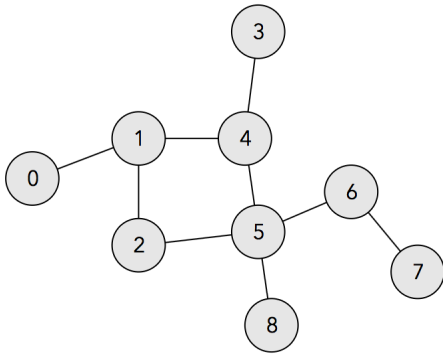


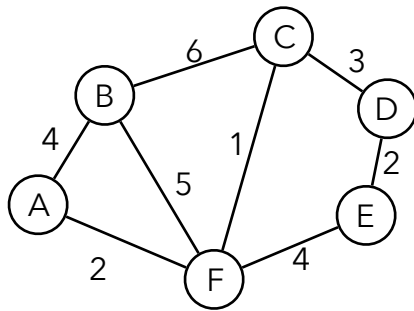
EECS 281, Week 7: Monday

Graphs

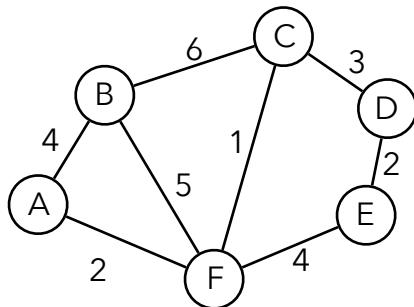
Write pre-order DFS, post-order DFS and BFS for this graph:



Prim's Algorithm



Kruskal's Algorithm



EECS 281, Week 7: Monday

Algorithm Design

Brute Force

Greedy

Divide and Conquer

Backtracking

Branch-and-bound

EECS 281, Week 7: Monday

Dynamic Programming

Fibonacci Sequence

Knapsack Problem

Huffman Coding

EECS 281, Week 7: Monday

Huff 'N Puff

Suppose we have an Encoder class (that uses Huffman coding) with this definition:

```
class Encoder {
public:
    // ...
private:
    // probabilities for each supported character, e.g. 'a'→0.2, 'b'→0.1
    unordered_map<char, double> probabilities;
    // encodings for each supported character, e.g. 'a'→"0", 'b'→"10"
    unordered_map<char, string> encodings;
    // ...
    void setEncodings();
};
```

Implement the `setEncodings` method. It's a private method that gets called during the initialization of the Encoder class, after the `probabilities` have been set but before `encodings` have been set. Hint: use `std::priority_queue`.

```
void Encoder::setEncodings() {

    auto cmp = [this](char a, char b) {
        return probabilities[a] < probabilities[b];
    };
    priority_queue<char, deque<char>, decltype(cmp)> characters(cmp);
    for (auto it = probabilities.begin(); it != probabilities.end(); ++it) {
        char character = it->first;
        characters.push(character);
    }
    string prefix = "";
    while (!characters.empty()) {
        char character = characters.top();
        characters.pop();
        string encoding = prefix + '1';
        encodings[character] = encoding;
        prefix += '0';
    }
}
```

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