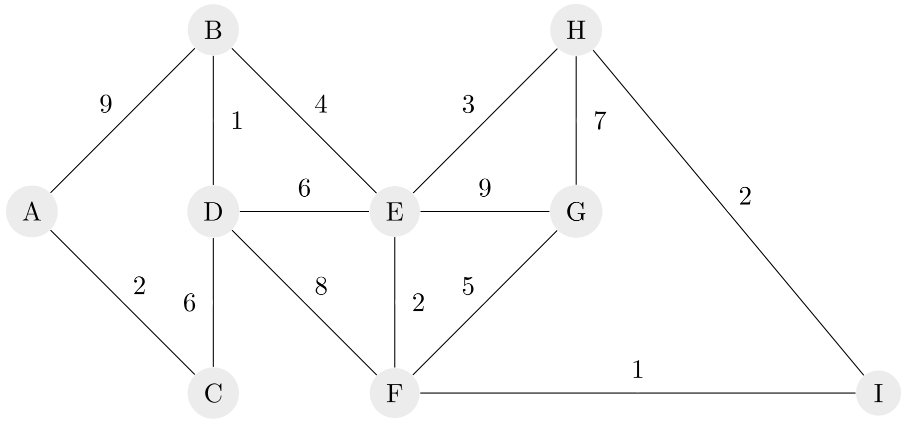
# CS61B Spring 2016
# Secret Section: Week 7

Tutor Team

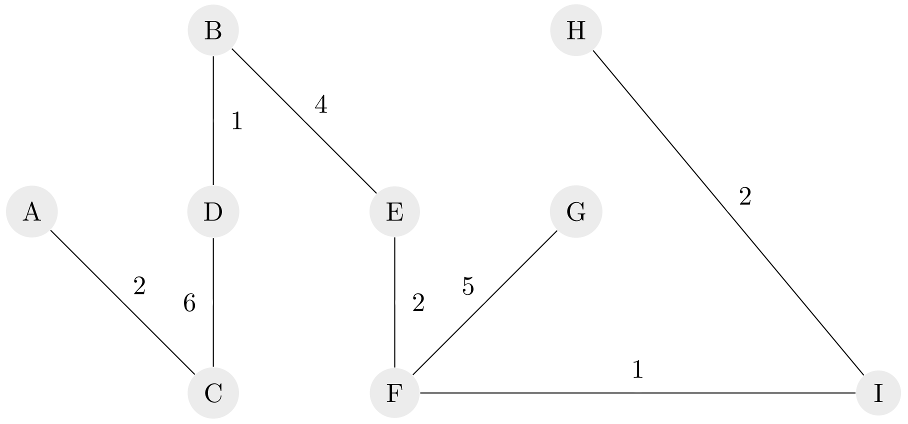
April 18, 2016

## 1 MST Algorithms

Consider the graph below.



Draw the MST for the graph above (there may be multiple answers).



In what order are the edges added by Prim's algorithm starting at vertex A?

**A-C, C-D, D-B, B-E, E-F, F-I, I-H, F-G**

What about Kruskal's algorithm?

**B-D, F-I, A-C, E-F, I-H, B-E, F-G, C-D**

## 2 Fast Fibonacci

Below is the standard recursive algorithm for calculating the n$^{th}$ fibonacci number:

```java
public int fib(int n) {
    if (n <= 1) {
        return n;
    } else {
        return fib(n-1) + fib(n-2);
    }
}
```

This algorithm is short and concise, but it's really slow! In fact, it runs in $\Theta(\phi^n)$ time. Lets try and develop a dynamic programming algorithm to solve it more efficiently.
The first thing to do is to determine what subproblems we need to solve. What are they?

**For fib(n), the subproblems are all fib(m) such that $m < n$.**

Let's try to visualize the problem as a DAG of recursive calls. Each node will be a call, and there should be an edge from fib(k) to fib(m) if fib(m) calls fib(k). Try an example with fib(5) below.

$$\begin{tikzcd}
\text{fib}(0) \arrow[rr, bend left] & \text{fib}(1) \arrow[r] \arrow[rr, bend right] & \text{fib}(2) \arrow[r] \arrow[rr, bend left] & \text{fib}(3) \arrow[r] \arrow[rr, bend right] & \text{fib}(4) \arrow[r] & \text{fib}(5)
\end{tikzcd}$$

Do you notice a pattern? What order could we efficiently solve these subproblems in so each problem's dependencies are solved before it?

**Increasing order by n.**

Using what we've seen so far, fill in the dynamic programming version of fibonacci below.

```java
public int fib(int n) {
    int[] results = new int[n+1];
    results[0] = 0;
    results[1] = 1;
    for (int i = 2; i <= n; i++) {
        results[i] = results[i-1] + results[i-2];
    }
    return results[n];
}
```

What is the time complexity of this new algorithm?

$$\Theta(n)$$

[BONUS]
Can you improve the algorithm above to use constant space? Try in the space below!

```java
public int fib(int n) {
    int first = 0;
    int second = 1;
    for (int i = 0; i < n; i++) {
        int sum = first + second;
        first = second;
        second = sum;
    }
    return first;
}
```