Meeting 3
10/11/15
11 AM – 2 PM

1. **Big O Complexity**

a. **O(1)**

Fastest runtime

```
i=0;
if(i==0)

        Do something();
}
while(i==0)
        ++i;
```

b. **O(Lg n)**

```
int count ones(int num)
{
        int count = 0;
        while(num)
        {
                count += num%2;
                num/=2;
        }
        return count;
}
```

c. **O(n)**

i. void loopThrough(int arr[])

```
{
        int i = 0;
        for(i=0; i < sizeof(arr)/sizeof(arr[0]); ++i)
                printf("%d", arr[i]);
        }
```

d. **O(nLgn)**

i. Nested O(n) and O(Lgn)

e. **O(n^2)**

i. 2 nested for loops

f. **O(2^n) / O(n!)**

i. Very slow

2. Puzzle

```
//Given an integer n, figure out if it’s a prime or not
    int isPrime(int num)
    {
        int i = 0;
        if (num == 1)
            return 0;
        else if(num == 0)
            return 0;
        else if(num == 2)
            return 1;
        else if(num % 2 == 0)
            return 0;
    else
    {
        for(i=3; i <= sqrt(num); i+=2)
        {
            if(num%i == 0)
                return 1;
        }
    }

//Running time: 0(n), Memory: O(1)
```