# Algorithm Analysis Treatise

Mike Maxim

September 11, 2002

## Big O Notation.

1.) $f(n) = 3n^2 + 2n + 1$
2.) $f(n) = 3\log n + n$
3.) $f(n) = \sum_{i=0}^{n} i$
4.) $f(n) = 4^{\log_2 n}$
5.) Given that $f(n) = n^{128}$ and $g(n) = 2^n$, using $\Omega$, $O$, and $\Theta$ notation, relate $f(n)$ and $g(n)$ two ways.

## Running Time Analysis.

1.) Write a recurrence relation for the running time of binary search as described in recitation. What is the Big-O solution for this recurrence?

```
void DoStuff(int[] array, int left, int right) {
    int mid = (left+right)/2;
    if (left < right)
        DoStuff(a,left,mid);
        DoStuff(a,mid,right);
        Combine(a,left,mid,right); (Runs in O(n) time)
    }
}
```

2.) What is a recurrence describing the running time of this block of code? What is the solution to the recurrence?

## Induction.

1.) Prove that the number of nodes in a perfect binary tree of height $H$ is $2^{H+1} - 1$.