Algorithm Analysis

Assignment 1 Order the following functions by growth rate: N, $\sqrt{N}$, $N^{1.5}$, N^2, Nlog N, N loglogN, $N\log^2 N$, $N\log(N^2)$, 2/N, 2^N,$2^{N/2}$, 37, $N^2\log N$, N^3. Indicate which functions grow at the same rate.

Assignment 2. Give an analysis of running time of each following code (Big-Oh)

```
1.Sum = 0;

for (i =1; i<= N; i++)  -> N
                                  O(N)
        Sum += i*i*i;    -> 3


2. Sum = 0;

for (i =1; i<= N; i++)     -> N
                                                O(N^3)
        for(j=0;j <= N*N; j++)   -> N^2

                Sum ++;


3. Sum = 0;

for (i =1; i<= N; i++)

        for(j=0;j <= i; j++)            O(N^2)

                Sum ++;


4. Sum = 0;

for (i =1; i<= N; i++)                     O(N^2)

        for(j=0;j <= i; j++)

                Sum ++;

5. Sum = 0;

for (i =1; i<= N; i++)     O(N)

        for(j=0;j <= i*i; j++)   O(N^2)
                                                  O(N^5)
                       k     k
                for(k=0;j <= j; j++)   O(N^2)

                        Sum ++;
```

6. Sum = 0;

```
for (i =1; i<= N; i++)    O(N)
        for(j=0;j <= i*i; j++)    O(N)
        if ( j%i== 0 )                          O(N^4)
                for(k=0;j <= j; j++)    O(N^2)
                        Sum ++;
```

7.

```
i=1;
s=1;
while( s<= N)     S = 1 + 2+ ... + k = N + 1= k(k+1) / 2
{
                                          k(k+1) / 2 = N + 1
                                          k^2 + k = 2(N+1)
i++;          while loop run k times      Then k = sqrt(N)
s+=i;
                                          O(sqrt(N))
}
```

Assignment 3, O(N^2)

```
for (int i = 0; i <= N; i++) {
        double term = 1.0;
        for (int j = 0; j < i; j++) {
            term *= X;
        }
        result += alpha[i] * term;
    }

    printf("F(X) = %lf\n", result);
```

Assignment 3, O(N)

```
double evaluate_polynomial(double x, int N) {
    double result = 0.0;
    for (int i = 0; i <= N; ++i) {
        result += coefficients[i] * pow(x, i);
    }
    return result;
}
```

Assignment 3 Write a program to evaluate the function $F(X) = \sum_{i=0}^{N} a_i X^i$. After that, calculate your running time?

b. If your running time is O(N^2), please find an algorithm with linear complexity.

Assignment 4.

a. Write a program to determine if a positive integer, N, is prime.
b. In terms of N, what is the worst-case running time of your program?

```
for (i = N; i>=1){
  i = i/2             k times
}
N -> N/2 -> N/4 -> ...

i = N / (2^k) = 1

 2^k = N
 k = log(N) / log(2)
```

```
#include <stdio.h>
#include <stdbool.h>
bool isPrime(long long int N){
  if (N <= 1){
    return false;
  }
  for (int i = 2; i < sqrt(N); i++){
    if (N % i == 0){
      return false;
    }
  }
  return true;
}
int main(){
  long long int N;
  scanf("%lld", &N);
  isPrime(N) ? printf("True") : printf("False");
  return 0;
}
```