# Homework 2

Casey Bates

19 April 2020

## 1 Multiplication and Division

**(a)**

$$O(log^2(N)): \qquad t = k \times n^2$$

It takes 3 nanoseconds to multiply 1000 bit numbers:

$$3 = k \times n^2$$

To find the time to multiply 5000 bit numbers, we must multiply n by a factor of 5:

$$k \times (5 \times n)^2 = 25 \times k \times n^2 \tag{1}$$

$$3 \times 25 = 75 \tag{2}$$

Therefore, it will take 75 nanoseconds to multiply 5000 bit numbers.

**(b)**

$$O(\sqrt{N}): \qquad t = k \times 2^{\frac{n}{2}}$$

It takes 11 nanoseconds to factor N using trial division:

$$11 = k \times 2^{\frac{n}{2}}$$

We know that N' is 10 bits larger than N:

$$k \times 2^{\frac{10+n}{2}} = k \times 2^5 \times 2^{\frac{n}{2}} = 32 \times k \times 2^{\frac{n}{2}} \tag{1}$$

$$11 \times 32 = 352 \tag{2}$$

Therefore, it will take 352 nanoseconds to factor N' using trial division

## 2 RSA

We know that the repeated squares algorithm used in RSA is $O(log^3(N))$

$$O(log^3(N)): \qquad t = k \times n^3$$

We must multiply our input by 4 to switch from 1024-bit RSA to 4096-bit RSA:

$$k \times (4 \times n)^3 = 64 \times k \times n^3 \tag{1}$$

$$64 \times k \times n^3 = 64 \times t \tag{2}$$

Therefore, 4096-bit RSA takes 64 times longer than 1024-bit RSA.

## 3 Summations

**(a)**
We know that a single addition will have a running time of $O(log_2(N))$
In the equation $(((1+2)+3)...+N)$ there will be $N-1$ additions.
Thus, the running time of this program will be:

$$O(log_2(N) \times (N-1)) = O(Nlog_2(N) - log_2(N))$$

**(b)**

Running time for addition by one: $O(1)$
Running time for multiplication: $O(log^2(N)$
Running time for division by two: $O(2^2) = O(4)$
So the total running time of $\frac{N(N-1)}{2}$ is:

$$O(1) + O(log^2(N)) + O(4) = O(5) + O(log^2(N)) \tag{1}$$

$$O(5) + O(log^2(N)) = O(log^2(N) + 5) \tag{2}$$

Since 5 is insignificant compared to $log^2(N)$ we can further simplify the equation.
Therefore, the running time is $O(log^2(N))$

## 4 $6^N$

A program to compute $6^N$ will have $N-1$ multiplications.
Since 6 has three bits, multiplication by 6 will take $O(3^2) = O(9)$
Thus, the running time of $6^N$ will be:

$$O(N-1) \times O(9) = O((9 \times N) - 9)$$

# 5 $\mathbf{X}^N$

A program to compute $X^N$ will have $N-1$ multiplications.
Multiplication by X will take $O(log^2(X))$
Thus, the running time of $X^N$ will be:

$$O(N-1) \times O(log^2(X)) = O(Nlog^2(X) - log^2(X))$$

# 6 Fibonacci

The equation $(((F_1 \times F_2) \times F_3)... \times F_n)$ will have $n-1$ multiplications.
Each number in the Fibonacci sequence increases by a factor of $\alpha$.
Multiplication by $\alpha$ takes $O(log^2(\alpha))$
Thus, the running time for the given Fibonacci algorithm is:

$$O(n-1) \times O(log^2(\alpha)) = O(n \times log^2(\alpha) - log^2(\alpha))$$