

Great Ideas in Complexity Analysis:

Why worst-case analysis?

Why Big-O notation?

Why polynomial time?

What is the running time of the following algorithm in terms of the input length?

```
def isPrime(n):
    if (n < 2):
        return False
    for factor in range(2,n):
        if (n % factor == 0):
            return False
    return True
```

Random-Access Machine (RAM) Model:

Small number:

Large number:

Examples:

```
def foo(int B):
    return B + B
```

```
def bar(string s):
    i = 0
    while (i < len(s)):
        print(s[i])
        i = i + 1
```

Are the numbers involved small or large?

Integer Addition

What is the running time?

```
def sum(A, B):  
    for i from 1 to B:  
        A += 1  
    return A
```

What is the running time of grade school algorithm?

Integer Multiplication

What is the running time of grade school algorithm?

Can we do better?

```
 $x = 5678$   
 $y = 1234$ 
```

Matrix Multiplication

Input:

Output:

Algorithm 1:

Algorithm 2:

Algorithm 3: