

A cartoon illustration of a classroom scene. On the left, a blackboard displays C++ code. On the right, a student with glasses is pointing at the board, and a teacher with a mustache is looking at the code. The teacher says, "NICE TRY." The code on the board is a loop that prints a sentence 500 times.

```
#include <iostream>
using namespace std;
int main( )
{
    int count;
    for(count=0; count < 500; count++)
        cout << "I will not throw paper airplanes in class." << endl;
    return 0;
}
```

WEND 10-3

NICE TRY.

*This is C++, but I bet you can still read it!

Announcements

- Hangman – due today
- Apt Set 2 – due Sept 16
 - Make ONE APT project
 - Add new class for each APT
 - Submit to Apt2
 - ALL APTs (even those from APT set 1)



Feedback

- Anonymous feedback
- UTA feedback
 - If someone is missing in the Link, let us know
 - If someone is GREAT, let us know

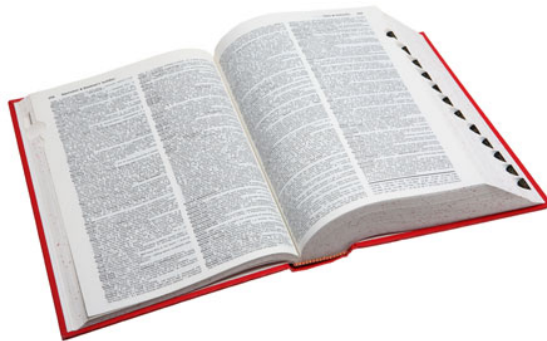


From last time

- Array – ordered, indexed, fixed length
- List – ordered, indexed, adjustable length
- Set – unordered, adjustable length, no doubles
- Map – unordered, pairs (key, value)

Map

- Unordered collection of values mapped to keys
 - dictionary
 - key – word
 - value - definition



Map

```
• Map<Double, Integer> map =  
    new HashMap<Double, Integer>();  
  
for(double d: map.keySet()){  
    System.out.println(d + ": " + map.get(d));  
}
```

<http://docs.oracle.com/javase/6/docs/api/java/util/HashMap.html>

Recitation

```
1  public class References {
2      public static void randomFunction2() {
3          String a = "Hello";
4          String b = "Goodbye";
5
6          a = b;
7          a = a.concat(" CS201");
8
9          System.out.println(a);
10         System.out.println(b);
11     }
12
13     public static void main(String[] args){
14         randomFunction2();
15     }
```

References

- <http://www.youtube.com/watch?v=vm5MNP7pn5>

Big-Oh

- Big-Oh
 - Estimate time required for a program
 - No units of time!!!!!!
 - Count operations



Big-Oh

- Assign costs to operations
 - Declarations cost 0 units
 - `double aDouble;`
 - Operations cost 1 unit
 - `aDouble = 4.56` //assignment
 - `aDouble * 5` //mathematical operation
 - `return aDouble;` //returns



Big-Oh

```
1 public double getArea(double r){  
2     double pi;  
3     pi = 3.14;  
4     double area;  
5     area = pi * r * r;  
6     return area;  
7 }
```



Big-Oh

- Assign costs to operations
 - Declarations cost 0 units
 - Operations cost 1 unit
- Write in Big-Oh notation

Big-Oh

```

1 public double getArea(double r){
2     double pi;           0
3     pi = 3.14;           1
4     double area;         0
5     area = pi * r * r;   3
6     return area;         1
7 }                        Total: 5

```

$O(5)$

Big-Oh

- Assign costs to operations
 - Declarations cost 0 units
 - Operations cost 1 unit
- Write in Big-Oh notation
- Simplify
 - Remove constants
 - $O(6) = O(1)$
 - $O(4N) = O(N)$
 - $O(3N^2 + 5) = O(N^2)$
 - Remove lower order terms
 - $O(N^2 + N) = O(N^2)$

Big-Oh

```

1 public double getArea(double r){
2     double pi;           0
3     pi = 3.14;           1
4     double area;         0
5     area = pi * r * r;   3
6     return area;         1
7 }                        Total: 5

```

$O(1)$

~~$O(5)$~~

Big-Oh

- Assign costs to operations
 - Declarations cost 0 units
 - Operations cost 1 unit
- Write in Big-Oh notation
- Simplify
 - Remove constants
 - $O(6) = O(1)$
 - $O(4N) = O(N)$
 - $O(3N^2 + 5) = O(N^2)$
 - Remove lower order terms
 - $O(N^2 + N) = O(N^2)$

Your turn

```

1 public static int sum( int n)
2 {
3     int partialSum;
4     partialSum = 0;
5     for(int i = 1; i <= n; i++)
6         partialSum += i * i * i;
7     return partialSum;
8 }

```

Your turn

```

1 public static int sum( int n)
2 {
3     int partialSum;           0
4     partialSum = 0;           1
5     for(int i = 1; i <= n; i++)  n
6         partialSum += i * i * i; 4
7     return partialSum;         1
8 }

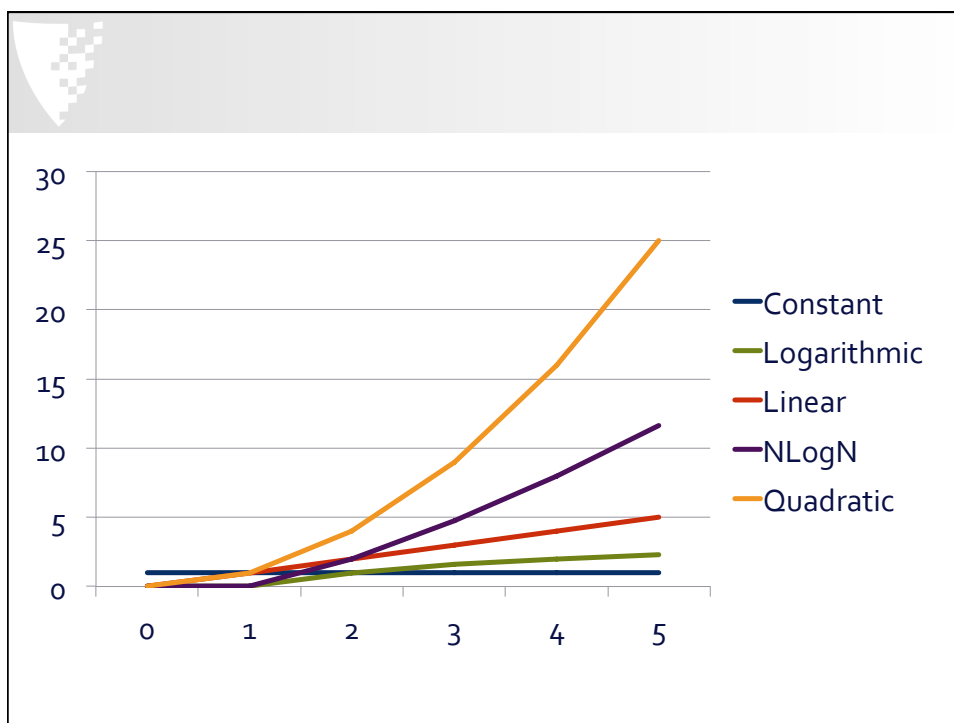
```

$$O(1 + N \cdot 4 + 1) = O(\cancel{4N} + \cancel{2})$$

Big-Oh

- Rules
 - for-loops
 - (statements in for-loop) * iterations
 - Nested for-loops (inside-out)
 - (statements in innermost for-loop) * iterations * iterations
 - Consecutive statements
 - Add them
 - If/else
 - Test + max(if, else)

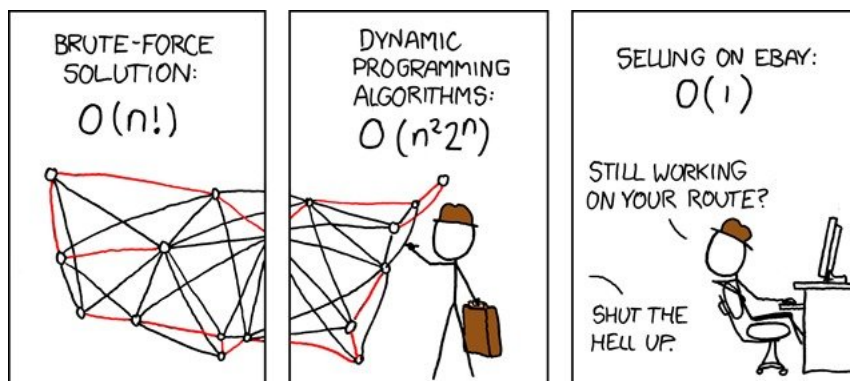
Function	Name
c	Constant
$\log N$	Logarithmic
$\log^2 N$	Log-squared
N	Linear
$N \log N$	
N^2	Quadratic
N^3	Cubic
2^N	Exponential



Practice

goo.gl/dGwqAL

The traveling salesperson





numberOne

```
1 public int numberOne(int n){  
2     return n;  
3 }
```



numberTwo

```
1 public int numberTwo(int n){  
2     int answer = 1;  
3     for(int i = 0; i < n; i++)  
4         answer *= n;  
5     return answer;  
6 }
```



numberThree

```
1 public int numberThree(int n){
2     int answer = 1;
3     for(int i = 0; i < n; i++)
4         for(int j = 0; j < n; j++)
5             answer *= n;
6     return answer;
7 }
```



numberFour

```
1 public int numberFour(int n){
2     int answer = 1;
3     for(int i = 0; i < n; i++)
4         answer *= n;
5     for(int i = 0; i < n; i++)
6         for(int j = 0; j < n; j++)
7             answer *= n;
8     return answer;
9 }
```



numberFive

```
1 public int numberFive(int n){  
2     int answer = 1;  
3     for(int i = 1; i <= n; i=i*2)  
4         answer *= n;  
5     return answer;  
}
```



numberSix

```
1 public int numberSix(int n){  
2     int answer = 1;  
3     for(int i = 1; i <= n; i=i*2)  
4         for(int j = 0; j < n; j++)  
5             answer *= n;  
6     return answer;  
7 }
```

numberSeven

```

1 public int numberSeven(int n){
2     if(numberTwo(n) > 10000){
3         return n;
4     }
5     else
6         return numberFive(n);
7 }

```

```

1 public int numberTwo(int n){
2     int answer = 1;
3     for(int i = 0; i < n; i++)
4         answer *= n;
5     return answer;
6 }

```

```

1 public int numberFive(int n){
2     int answer = 1;
3     for(int i = 1; i <= n; i=i*2)
4         answer *= n;
5     return answer;

```

Announcements

- Hangman – due today
- Apt Set 2 – due Sept 16
 - Make ONE APT project
 - Add new class for each APT
 - Submit to Apt2
 - ALL APTs (even those from APT set 1)