Today

- Big-Oh

- Snarf today's code

Announcements

- Hangman – Due tomorrow!
- APT set 2 – Due Jan 29
- NO recitation prep for Friday
  - Bring your book (if you have one)

- UTAs are in the LINK!!!!!!!!

- Data Structures
  - Array
  - ArrayList
  - HashSet
  - HashMap

- When to use?
  - function and time

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

Big-Oh

- How to calculate Big-Oh
  - Assign costs to operations

Big-Oh

- Declarations cost 0 units
  - double d;
- Operations cost 1 unit
  - d = 4.56 //assignment
  - d * 5 //mathematical operation
  - return d; //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

```
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
```

Big-Oh

- How to calculate Big-Oh
  - Assign costs to operations
  - 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

- How to calculate Big-Oh
  - Assign costs to operations
  - Write in Big-Oh notation
  - Simplify

Big-Oh

- Simplify
  - Remove constants!
    - $O(4N) = O(N)$
    - $O(3N^2 + 5) = O(N^2)$
  - 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(5) = O(1)

Big-Oh

- How to calculate Big-Oh
  - Assign costs to operations
  - Write in Big-Oh notation
  - Simplify

Code

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

Code

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

Code

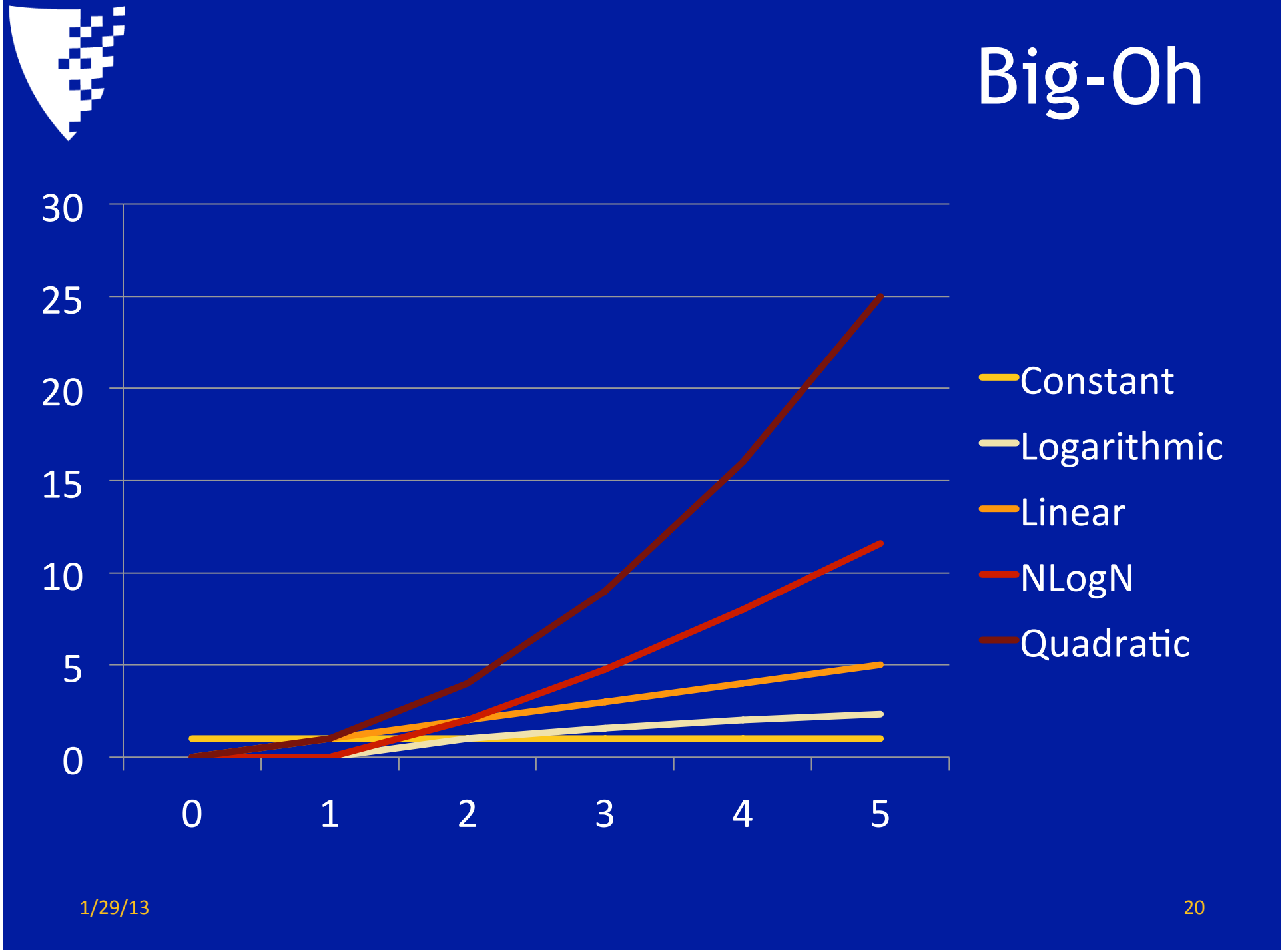
```
1  public static int sum( int n)
2  {
3       int partialSum;                  0
4       particalSum = 0;                 1
5       for(int i = 1; i <= n; i++)      n
6            particalSum += i * i * i;      4
7       return partialSum;               1
8  }
```

O(1+N*4 + 1) = O(4N+2) = O(N)

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



Big-Oh

30
25
20
15
10
5
0
0
1
2
3
4
5
Constant
Logarithmic
Linear
NLogN
Quadratic

Rules

- $T_1(N) = O(f(N))$ and $T_2(N) = O(g(N))$
  - $T_1(N) + T_2(N) = O(f(N) + g(N))$
  - $T_1(N) * T_2(N) = O(f(N) * g(N))$

- $\log^k N = O(N)$

Rules

- Remove constants!
  - $O(4N) = O(N)$
  - $O(3N^2 + 5) = O(N^2)$
- Remove lower order terms
  - $O(N^2 + N) = O(N^2)$

Practice

- Snarf today's code
- Complete the form
  - http://goo.gl/kuuHM
- For the timings
  - Choose an N that is DIFFERENT from neighbors

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

Announcements

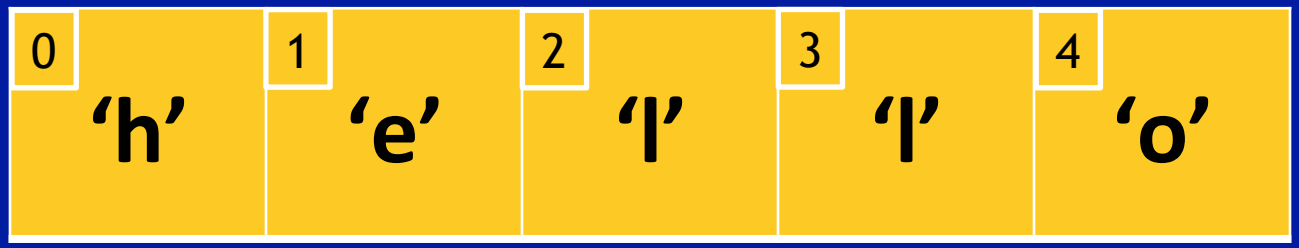
- Hangman – Due tomorrow!
- APT set 2 – Due Jan 29
- NO recitation prep for Friday
  - Bring your book (if you have one)

- UTAs are in the LINK!!!!!!!!

Data Structure	Add	Contains
Array	O(1)	O(N)
ArrayList	O(1)	O(N)
HashSet	O(1)	O(1)

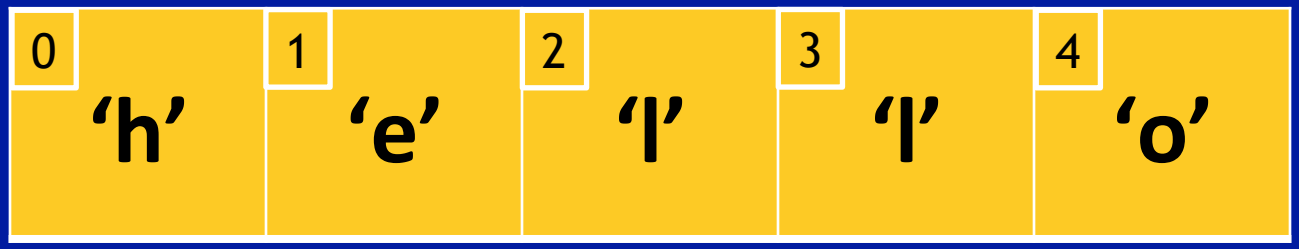
Contains

- Array and ArrayList
  - Best case?
  - Average case?
  - Worst case?

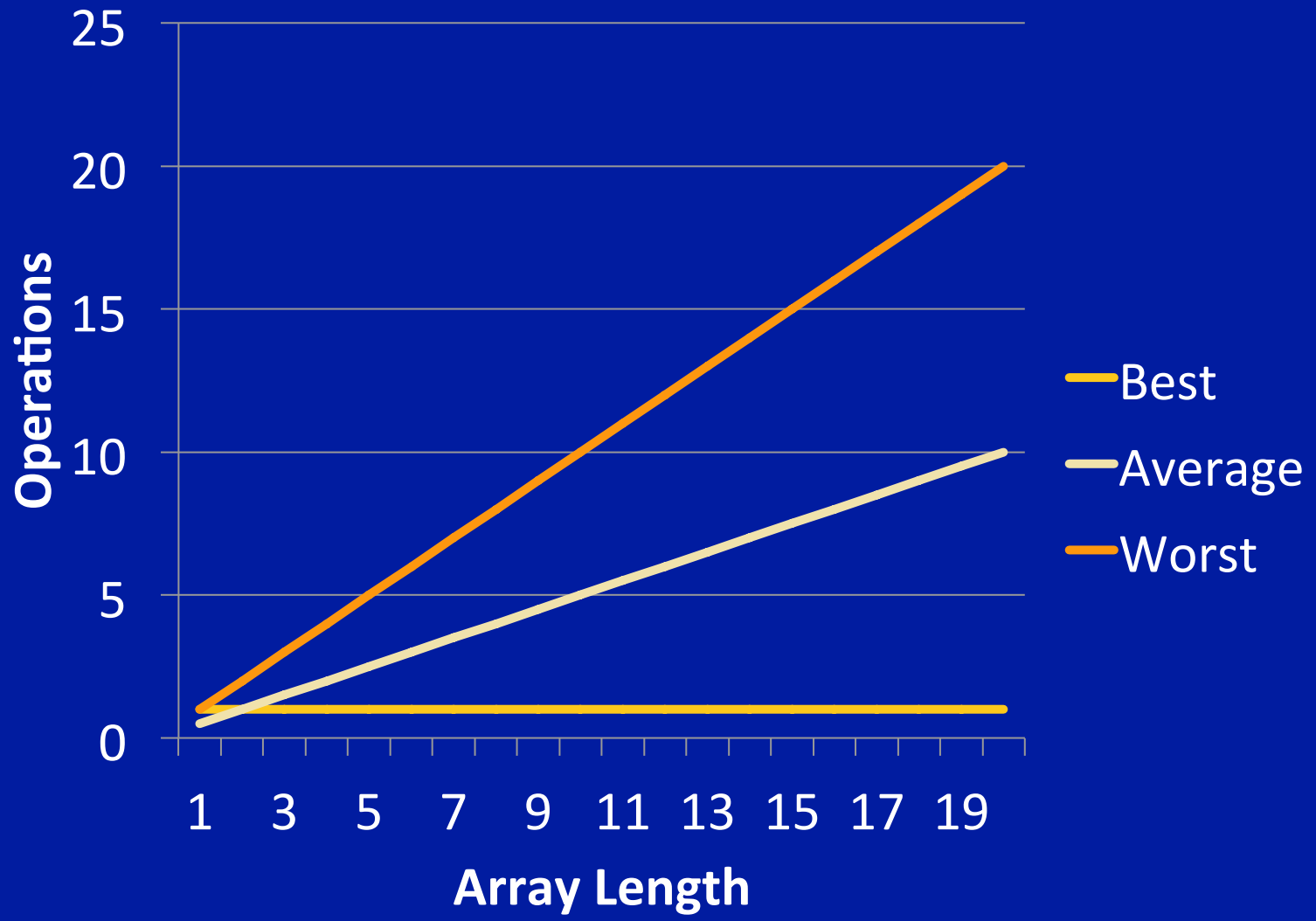


Contains

- Array and ArrayList
  - Best case? 1
  - Average case? N/2
  - Worst case? N



Contains


25
20
15
10
5
0
Operations
1
3
5
7
9
11
13
15
17
19
Array Length
Best
Average
Worst