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
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  }

public double getArea(double r) {
    double pi;
    pi = 3.14;
    double area;
    area = pi * r * r;
    return area;
}
Big-Oh

• How to calculate Big-Oh
  • Assign costs to operations
  • Write in Big-Oh notation
public double getArea(double r){
    double pi;
    pi = 3.14;
    double area;
    area = pi * r * r;
    return area;
}

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
    - \( O(N^2 + N) = O(N^2) \)
public double getArea(double r) {
    double pi = 3.14;
    double area = pi * r * r;
    return area;
}

O(5) = O(1)
Big-Oh

- How to calculate Big-Oh
  - Assign costs to operations
  - Write in Big-Oh notation
  - Simplify
public static int sum(int n) {
    int partialSum;
    partialSum = 0;
    for(int i = 1; i <= n; i++)
        partialSum += i * i * i;
    return partialSum;
}
public static int sum(int n) {
    int partialSum = 0;
    for (int i = 1; i <= n; i++) {
        partialSum += i * i * i;
    }
    return partialSum;
}
public static int sum(int n) {
    int partialSum; 0
    partialSum = 0; 1
    for(int i = 1; i <= n; i++) n
        partialSum += i * i * i; 4
    return partialSum; 1
}

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)
<table>
<thead>
<tr>
<th>Function</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>Constant</td>
</tr>
<tr>
<td>$\log N$</td>
<td>Logarithmic</td>
</tr>
<tr>
<td>$\log^2 N$</td>
<td>Log-squared</td>
</tr>
<tr>
<td>$N$</td>
<td>Linear</td>
</tr>
<tr>
<td>$N \log N$</td>
<td></td>
</tr>
<tr>
<td>$N^2$</td>
<td>Quadratic</td>
</tr>
<tr>
<td>$N^3$</td>
<td>Cubic</td>
</tr>
<tr>
<td>$2^N$</td>
<td>Exponential</td>
</tr>
</tbody>
</table>
Rules

- $T_1(N) = O(f(N))$ and $T_2(N) = O(g(N))$
- $T_1(N) + T_2(N) = O(f(N) + g(N))$
- $T_1(N) \times T_2(N) = O(f(N) \times g(N))$
- $\log^k N = O(N)$
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!!!!!!!!!!
<table>
<thead>
<tr>
<th>Data Structure</th>
<th>Add</th>
<th>Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td>O(1)</td>
<td>O(N)</td>
</tr>
<tr>
<td>ArrayList</td>
<td>O(1)</td>
<td>O(N)</td>
</tr>
<tr>
<td>HashSet</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
</tbody>
</table>
Contains

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

• Array and ArrayList

• Best case? 1
• Average case? N/2
• Worst case? N