Science 1000: Lecture #3 (Wareham):

Necessary Lies:
Asymptotic Worst-case
Time Complexity Analysis

Comparing running times is hard?
Not really.
Comparing Algorithms: What’s Best?

- Best algorithm = algorithm with lowest running time.
- Comparing algorithms by raw running time problematic:
  - Raw running times machine / language / OS dependent.
  - Raw running times input dependent.
  - Algorithm may not be implemented in a program.

HOW DO WE MEASURE ALGORITHM RUNNING TIME?
Necessary Lie #1: Runtime Equivalence of Instructions

- Compute runtime on an input by counting the number of instructions executed.
- Is machine-independent (raw abstract runtime).
Necessary Lie #2: Worst-Case Runtime Summary

- Group inputs by input size; summarize each size by largest runtime for that size.
- Is input-independent (worst-case time complexity).
Necessary Lie #3:
Asymptotic Smoothing

- Reduce time complexity function to largest term.
- Is simple (asymptotic worst-case time complexity).
Deriving Worst-Case Time Complexities

- If already have time complexity, select largest term, e.g.,

\[ 2 \log n + 4 \implies O(\log n) \]

\[ 3n^2 + 1000n + 13 \implies O(n^2) \]

\[ 12n^4 + 5n^2 + 900 \implies O(n^4) \]

\[ (3 \times 2^n) + 900n^{50} + 57 \implies O(2^n) \]
Deriving Worst-Case Time Complexities (Cont’d)

- Otherwise, multiply out “deepest” loop-chain in algorithm, e.g., \( n \times n = O(n^2) \) time for List Sort.

```plaintext
for i = 1 to n - 1 do
    min_pos = i
    for scan = i + 1 to n do
        if (L[scan] < L[min_pos]) then
            min_pos = scan
    temp = L[min_pos]
    L[min_pos] = L[i]
    L[i] = temp
```
Time Complexity Magnitudes

\[ O(\log n) \quad \text{Logarithmic Time} \quad \text{(Binary Search)} \]

\[ O(n) \quad \text{Linear Time} \quad \text{(Linear Search)} \]

\[ O(n^2) \quad \text{Quadratic Time} \quad \text{(List Sort)} \]

\[ O(2^n) \quad \text{Exponential Time} \quad \text{(Bin Packing)} \]

**Polynomial Time** = \( O(n^c) \) time for constant \( c \)
# Table of Doom (1 Gigaflop/s Version)

<table>
<thead>
<tr>
<th>Input Size ($n$)</th>
<th>B-Search ($\log_2 n$)</th>
<th>L-Search ($n$)</th>
<th>Sort ($n^2$)</th>
<th>MST ($n^3$)</th>
<th>BP-E ($2^n$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
</tr>
<tr>
<td>50</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>13 days</td>
</tr>
<tr>
<td>100</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>$4 \times 10^{13}$ years</td>
</tr>
<tr>
<td>1000</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>1 second</td>
<td>$4 \times 10^{284}$ years</td>
</tr>
<tr>
<td>one million</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>2 minutes</td>
<td>30 years</td>
<td>–</td>
</tr>
<tr>
<td>300 million</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
<td>10 days</td>
<td>$9 \times 10^5$ years</td>
<td>–</td>
</tr>
<tr>
<td>five billion</td>
<td>&lt; 1 second</td>
<td>5 seconds</td>
<td>8 centuries</td>
<td>$4 \times 10^{12}$ years</td>
<td>–</td>
</tr>
</tbody>
</table>
Science 1000: Lecture #3 (Wareham):

Necessary Lies:
Asymptotic Worst-case Time Complexity Analysis

Comparing running times is hard?
Not really.