CSC 1052 – Algorithms & Data Structures II: Complexity

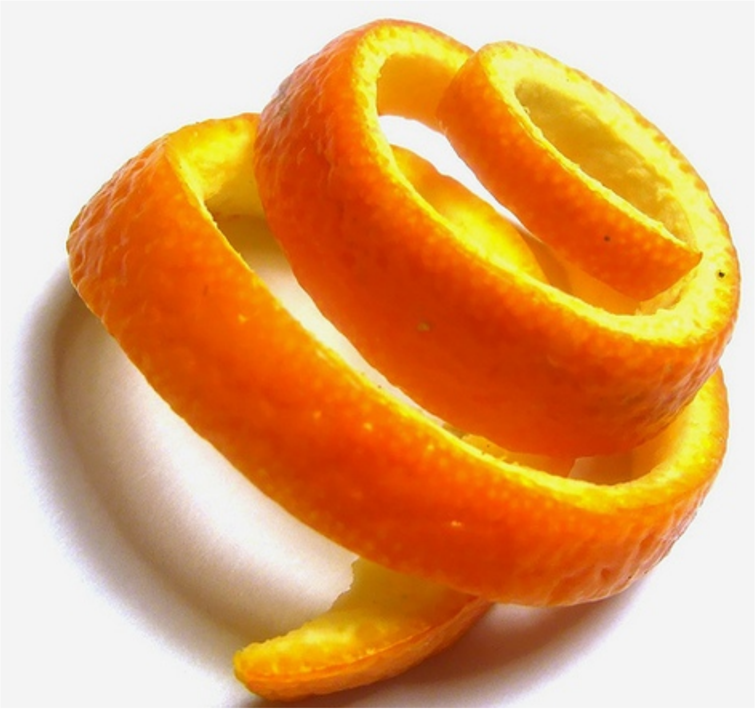
Professor Henry Carter
Spring 2017

Recap

- Good programming requires careful practice
  - Organize your code
  - Throw useful exceptions and errors
  - Comment your code
- Data structures allow us to arrange data for fast and easy access
- The arrangement of data in memory greatly impacts the way Java handles the data

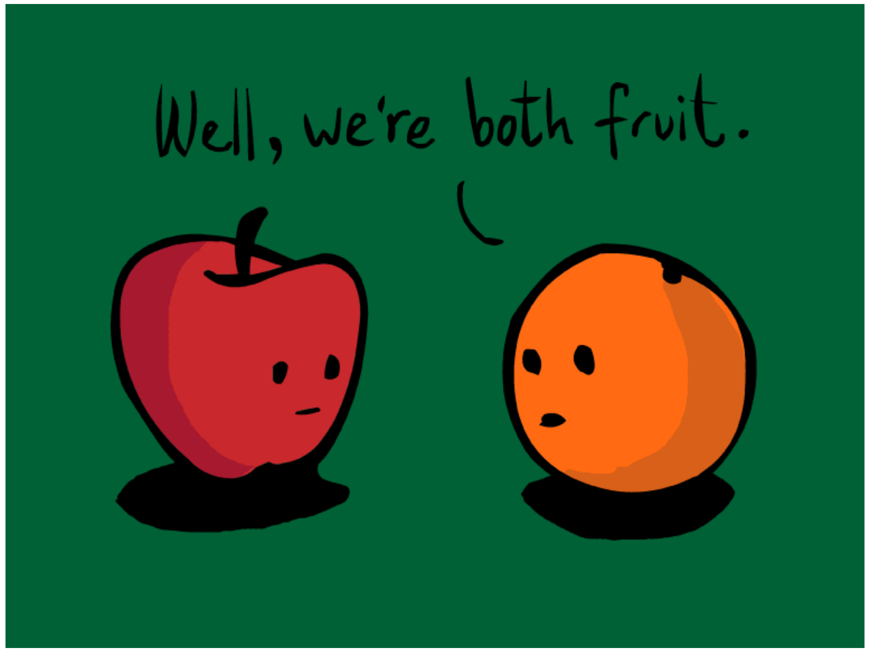
Data Structures

- A lot of what we do will be describing data structures behavior
- For implementation independent data structures, there will be many ways to "Peel the orange"
- Which way is best?



The Language of Efficiency

- We need a way to compare different implementations
- Correctness of the implementation
- How long does it take?
- How much space does it take?



Sequential Search

Problem: guess secret number between 1 and 1,000 (the Hi-Lo game)

```
Hi-Lo Sequential Search:
  Set guess to 0
  do
    Increment guess by 1
    Announce guess
  while (guess is not
           correct)
```

Sequential Search: Operations

Case-wise analysis

- Best case
- Average case
- Worst case

Binary Search

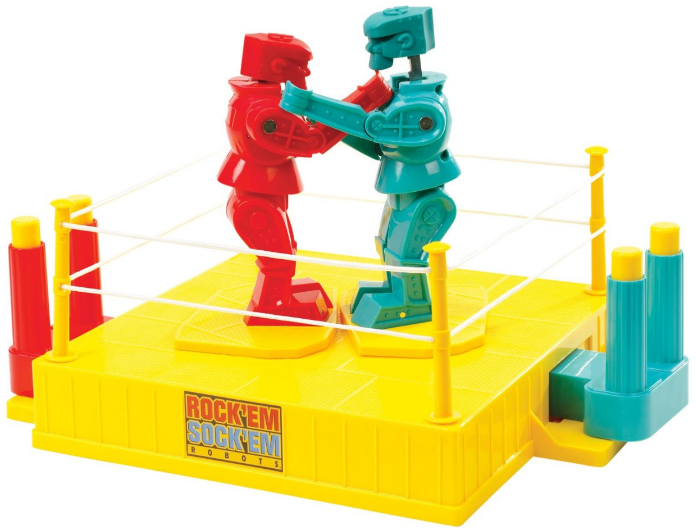
Problem: guess secret number between 1 and 1,000 (the Hi-Lo game)

```
Hi-Lo Sequential Search:
   Set range to 1…1000
   do
      Set guess to middle of range
      Announce guess
      If(guess too high)
         Set range to first half
      If(guess too low)
         Set range to second half
   while (guess is not correct)
```

Binary Search

Sequential Search vs. Binary Search

- Sequential search: $2n + 1$
- Binary search: $5 \log n + 3$
- Which do you prefer?



Operation Counts

Size	Sequential Search	Binary Search
N	2N + 1	5 log N + 3
2	5	8
4	9	13
8	17	18
16	33	23
32	65	28
1024	2049	53

Order of growth

- Constant multiples have less impact as N grows
- Approximate the efficiency with an order of growth
- These orders of growth are typically easier to calculate and compare

Simplified orders of growth

Size (N)	Linear (N)	Logarithmic (log N)
2	2	1
4	4	2
8	8	3
16	16	4
32	32	5
1024	1024	10
1,000,000	1,000,000	20

Common Orders of Growth

- $O(1)$ "constant"
- $O(\log_2 N)$ "logarithmic"
- $O(N)$ "linear"
- $O(N \log_2 N)$ "N log N" or "linearithmic"
- $O(N^2)$ "quadratic"
- $O(2^N)$ "exponential"

Determining the order of growth

- What are the added terms of the op count?
- Which term is growing the fastest?
- What function is being applied to N?

Example functions

- $N^2 + 3N$
- $1000\ N^{3\ +}\ N^5 + 268$
- $40 \log N + N \log N$

Selection Sort

SelectionSort

for current going from 0 to SIZE - 2

  Find the index in the array of the smallest unsorted element

  Swap the current element with the smallest unsorted one

Example Selection Sort

- 3 5 8 6 1 7

Selection Sort: Op Count

Example Comparison Values

N	$\log_2 N$	$N\log_2 N$	N^2	N^3	2^N
1	0	1	1	1	2
2	1	2	4	8	4
4	2	8	16	64	16
16	4	64	256	4,096	65,536
64	6	384	4,096	262,144	**requires 20 digits**
128	7	896	16,384	2,097,152	**requires 39 digits**
256	8	2,048	65,536	16,777,216	**requires 78 digits**

Quick Tips

- Look at the loops
- Look for division
- Think about best case vs worst case

Exercises

- Identify the order of growth for each of the following code segments:

- ```
  count = 0;
  for (i = 1; i <= N; i++)
      for (j = 1; j <= N; j++)
          count++;
  ```

- ```
  count = 0;
  for (i = 1; i <= N; i++)
      count++;
  for (i = 1; i <= N; i++)
          count++;
  ```

- ```
  count = 0;
  for (i = 1; i <= N; i++)
      for (j = 1; j <= 10; j++)
          count++;
  ```

- ```
  count = 0;
  value = N;
  value = N*(N-1)
  count = count + value
  ```

Recap

- Evaluating efficiency means many different things depending on context
- Time complexity is a common metric based on the number of operations with respect to input size
- Orders of growth allow for a simplified way to compare time complexity of different algorithms and implementations

Next Time...

- Dale, Joyce, Weems Chapter 2.1-2.3
  - Remember, you need to read it BEFORE you come to class!
- Check the course webpage for practice problems
- Peer Tutors
  - http://www.csc.villanova.edu/help/