

Sorting, selecting and complexity

Prof. Ramin Zabih

<http://cs100r.cs.cornell.edu>



Cornell University
Computer Science

Administrivia

- Assignment 1 is due tomorrow 5PM
 - Lots of TA time available (check the web)
 - Robot shortage will end soon...



Consider our algorithm

- Given an array of n numbers, find the number smaller than exactly m of them
- Strategy:
 - Remove the biggest number
 - Do this m times
 - The answer is the biggest remaining number



Performance of our algorithm

- What is the worst case?
 - $m = n/2$
 - a.k.a., finding the median!
- Suppose our input has n points
 - Roughly how much work do we need to do, as a function of n , in the worst case?
 - Examine n elements to find the biggest
 - Examine $n-1$ elements to find the biggest
 - ...
 - Do this $n/2$ times



How much work is this?

- No matter what computer we use, we have to do something $n/2$ times
 - Namely, find the biggest element in an array
 - The array is at least $n/2$ elements long
 - Usually longer
 - Can we find the biggest element without looking at all of them?
- The work grows as roughly n^2
 - Suppose we add a new number to the array
 - The extra work is proportional to how many numbers we already have



Classes of algorithm speed

- Constant time algorithms, $O(1)$
 - Do not depend on the input
 - Example: find the first element
- Linear time algorithms, $O(n)$
 - Constant amount of work for every input item
 - In the worst case, as always
 - Example: find the largest element
- Quadratic time algorithms, $O(n^2)$
 - Linear amount of work for every input item
 - Example: dumb trimming method



Asymptotic analysis picture

- If you plot n versus running time you get:
 - A flat line for $O(1)$
 - A sloped line for $O(n)$
 - A parabola for $O(n^2)$
- Different hardware only affects the parameters (i.e., line slope)
- As n gets big, the “dumber” algorithm by this measure always loses eventually



Limitations of this approach

- Obviously, the worst case doesn't always happen in practice
 - But this closely tracks real performance!
 - It's so good that the exceptions are famous
- It's **much** harder to reason about the average case
 - You need a definition of average that really matches your situation
 - Average case analysis doesn't track real performance nearly as well

