# MA/CSSE 473 Day 12

**Amortization (growable Array)**

**Knuth interview**

**Brute Force Examples**

## MA/CSSE 473 Day 12

- A6 Due date postponement:
  - You have been working *very* hard in this course.
  - Perhaps it's time for a little break.
  - A6 will be due next Tuesday, rather than this Friday.
  - (If you think you'll be busier next week than this week, you can go ahead and get A6 out of the way this week)
- I have another medical appointment in Indianapolis this afternoon. I will be gone after 5th period.
- I expect to be available most of tomorrow; meetings 10:50-11:40, 2:30-3:00, 4:30-5:30

Q1-2

# Amortized efficiency analysis

- P49-50 in the textbook
- We analyze not just a single operation, but a sequence of operations performed on the same structure
  - We conclude something about the worst-case of the average of all of the operations in the sequence
- Example: Growable array exercise from 220/230, which we will quickly review today

Q3

# Growable Array (implement ArrayList)

- An ArrayList has a *size* and a *capacity*
- The capacity is the length of the fixed-size array currently allocated to hold the list elements
- For definiteness, we start with *size*=0 and *capacity*=12
- We add a total of N items (N is not known in advance), one a t a time, each to the end of the structure
- When there is no room in the array (i.e. capacity=size and we need to add another element)
  - Allocate a new, larger array
  - copy the *size* existing elements to the new array
  - add the new element to the new array
- What is the total/average overhead (due to element copying) if
  a. we add one to the array capacity each time we have to grow it?
  b. we double the array capacity each time we have to grow it?
- Note in the second case that the amortized worst-case cost is asymptotically less than the worst case for a single element
- Every time we have to enlarge the capacity, we make it so we do not have to enlarge again soon

Q4

## Brute Force Algorithms

- Straightforward, simple, not subtle, usually a simple application of the problem definition.
- Often not very efficient
- Easy to implement, so often the best choice if you know you'll only apply it to small input sizes

- Take 10 minutes to do quiz questions 5-6, (and 1-2 if you did not already answer them)

Q5-6

## Donald Knuth Interview

- List a few things that you found interesting in the interview

- What questions would you ask Donald Knuth if you had the chance?

# What is a brute force approach to

1. Calculate the $n^{th}$ Fibonacci number?
2. Compute the $n^{th}$ power of an integer?
3. Search for a particular value in a sorted array?
4. Sort an array?
5. Search for a substring of a string?
6. Find the maximum contiguous subsequence? in an array of integers?
7. Find the largest element in a Binary Search Tree?
8. Find the two closest points among N points in the plane?
9. Find the convex hull of a set of points in the plane?
10. Find the shortest path from vertex A to vertex B in a weighted graph?
11. Solve the traveling salesman problem?
12. Solve the knapsack problem?
13. Solve the assignment problem?
14. Solve the n×n non-attacking chess queens problem?
15. Other problems that you can think of?