Recursion 1

CS 146 - Spring 2017

Review question

Grade-school multiplication algorithm
is said to run in quadratic time.

What does that mean?
Quadratic as a function of what variable?

Today

- Basic recursion
- Admin stuff: I will collect your pre-reqs documents

Last time: factorization

- Input: a composite (non-prime) integer
- Output: one non-trivial solution

- Give an algorithm that solves this problem.
- How many division operations?
- Is this a linear-time algorithm? Why?

Prime factorization

- Input: an integer
- Output: a list of all its prime factors

- Give an algorithm which solves this problem using the solution of the previous problem as a building block

Reusing known solutions

- Common technique for problem solving
- Idea: if you don't know how to solve a problem, try changing your problem into something you can solve (or that has been solved by someone else)
- reusable known solutions, aka
  - Software: **API**
  - Algorithms: **black box**
  - simply: **building blocks**

Recursion

- Uses itself as a building block to solve itself!
- How??!

Examples and practice

Download the code at:

https://github.com/jnylam/SJSU-cs146-s17

Pre-req documents

- On transcripts, highlight or underline
  - your name
  - your enrollment status
  - Math 42 or discrete math
  - Math 30 or calculus 1
  - CS 49 J or Java
  - CS 46 B or intro to data structures
- Community college: highlight courses on equivalency page
- Graduating seniors: have yellow card ready

Please staple!

Recap: thinking recursively

- don’t solve the whole problem
- just do a little bit of work to make progress
- to construct a solution, we take the result of the recursive call and use it to build the solution for the current call