Unfinished Lecture 2: Friday, September 7, 2001

today

- analysis cont'd
  - analysis using RAM
  - what Θ means (informally)
  - asymptotic running time
  - worst/average/best case
  - insertion sort analysis
- correctness
  - loop invariants
  - insertion sort correctness

analysis of RAM programs

- time: instructions executed
- space: memory locations used

- example: running time to multiply a*b?
- answer **depends on size of a,b**
  - if a,b,a*b each fit into one RAM memory word
    then constant number of RAM instr'ns, so constant time
  - if not
    need multiple words ro represent a,b,a*b
    can show number RAM instr'ns proportional to
    (words to represent a)*(words to represent b)
  - with k bits per word, number x needs using $\lg(x)/k$ words
  - time proportional to $\lg(a) \times \lg(b)$ k constant
  - write $\Theta(\ \lg(a) \times \lg(b)\ )$ time
- **log cost RAM**
  - assume numbers may not fit into one memory word
  - e.g. $x + y$ takes $\Theta(\ \lg(x) + \lg(y)\ )$ time and space
- **uniform cost RAM model**
  - assume that each number fits into one memory word
  - $x + y$ takes $\Theta(1)$ time and space
- unless otherwise stated, assume uniform cost RAM ☺

running time analysis (cont'd)

- model of computation: RAM
- problem: running time varies with input
- idea
  - estimate worst/average/best performance
  - make estimate a *function* of input *size*
  - e.g. sorting: size usually number of numbers
  - guarantee of performance ☺

kinds of analysis

- worst case
  - $T(n)$ max time over all inputs of size n
- average case
  - must specify input distribution over which average computed
  - most common: assume uniform (a.k.a. equiprobable) input distribution (all inputs of size n equally likely)
  - useful but usually difficult
    e.g. [Hayward/McDiarmid: avg. case heap construction]
- best case
  - useful for lower bounds
  - not otherwise useful: any alg'm can be modified to have fast best case
    (add: if input is [particular case] then return [particular answer])