CSC236 fall 2012

correct after & before

Danny Heap
heap@cs.toronto.edu
BA4270 (behind elevators)
http://www.cdf.toronto.edu/~heap/236/F12/
416-978-5899

Using Introduction to the Theory of Computation, Chapter 2

Last week's correctness proof of recBinSearch.
- weaken the pre-
- A not necessarily sorted.

post: return $0 \le p \le n$
$b < p \Rightarrow A[p-1] < x$
$p \le e \Rightarrow x \le A[p]$

Computer Science
UNIVERSITY OF TORONTO

Outline

power

notes

integer power

```
def power(x, y) :
  z = 1
  m = 0
  while m < y :
    z = z * x
    m = m + 1
  return z
```

- precondition?
- postcondition?
- notation for mutation

partial correctness

precondition+execution+termination imply postcondition

a loop invariant helps get us closer

partial correctness

precondition+execution+termination imply postcondition

a loop invariant helps get us closer

prove partial correctness

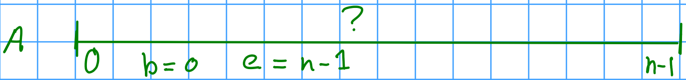
prove termination

associate a decreasing sequence in $\mathbb{N}$ with loop iterations

it helps to add claims to the loop invariant

put it together — correctness

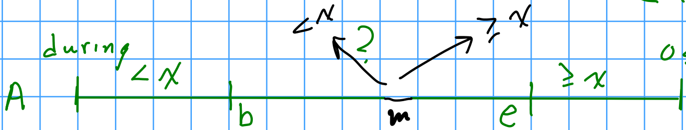
pre: A is sorted ↑ and x is comparable elts of A



post $0 \le p \le n \wedge A[0..p-1] < x \le A[p..n-1]$

$\langle e_i + 1 - b_i \rangle \in \mathbb{N}$

$0 \le b \le e \le n-1$



LI $A[0..b-1] < x \le A[e+1..n-1] \wedge 0 \le b \le e+1$

```
i=0 →   while b ≤ e :
            m = (b + e) // 2
            if A[m] < x :
                b = m + 1
            else:
                e = m - 1
        # out of loop
        return b
```

$blah_i$ is the state of blah at end of ith iteration

"derive" conditions from pictures

LI $A[0..b-1] < x \leq A[e+1..n-1]$ # suppress
$\wedge\ b \leq e+1$ # subscript
here.

Show $\langle e_i + 1 - b_i \rangle > \langle e_{i+1} + 1 - b_{i+1} \rangle$

No $m_{i+1} = (e_i + b_i) // 2$

Case 1 $b_{i+1} = m_{i+1} + 1 \wedge e_{i+1} = e_i$

Case 2 $e_{i+1} = m_{i+1} - 1,\ b_{i+1} = b_i$

$b_i \leq m_{i+1} \leq e_i$

do we have termination?

notes