

# Discrete Mathematical Structures

## CS 3233 Lecture Fifteen

Prof. William Winsborough

September 23, 2005

# Assignment 5

- Due Thursday, October 13, 2pm, at Dr. Winsborough's office
- Section 2.1
  - 12, 24, 26, 42
  - In exercise 24, you may assume that the following set-iterator operation exists and is available for your use:
    - For each <variable> in <set> do <statement block>
- Material from Chapter 2 will not be on the midterm 10/7

# Business

- Class schedule 9/26 – 10/7/2005:
  - No lecture next week (9/26-30)
  - Meet with TA during lecture time ALL the following week (10/3-7)
    - 10/3 and 10/5: Review of a mock exam that I will post by Sunday for you to use in preparing for the mid-term
    - 10/7: Midterm
  - Recitation meets both weeks I am gone
  - I will be back 10/10/2005
- Questions??
- Outline for today
  - Short further discussion of intuition about how there can be more reals than natural numbers
  - Start section 2.1
  - Discuss any questions about material that students wish to discuss, time permitting

# Intuitive Comment about Comparison of $\mathbb{N}$ and $\mathbb{R}$

- The representation of a real is infinite, while the representation of each natural number is finite
  - A real is an infinite sequence of decimals
  - A natural is constructed from 0 by applying successor (+1) a finite (but unbounded) number of times
  - Since each real has more information in it, it makes sense that there would be more of them

# Algorithms

- Read section 2.1
- Definition
  - An *algorithm* is a finite collection of precise instructions for performing a computation to solve a problem
  - Input and output values are elements of specified sets
- Desirable characteristics:
  - Definiteness: Steps are precisely defined
    - To be really precise, must use a formal *computational model*, such as a Turing machine or the lambda calculus
  - Effectiveness: It must be possible to perform each step using a bounded amount of time and storage space
    - Bounded means there is an amount of time that is always sufficient
  - Correctness, Termination, Generality

# Examples

- Searching: Determine whether a given value is contained in an input sequence of integers

# Linear Search

```
proc linear search(x:int; a1,a2,...,an: distinct ints)
i := 1
while (i ≤ n and x ≠ ai)
    i := i + 1
if i ≤ n then location := i else location := 0
```

- How many comparisons are performed?

# Binary Search

**proc** binary search(x: int; a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>n</sub>: increasing ints)

i:=1 {left end of search interval}

j:=n {right end of search interval}

**while** i<j **begin**

    m:=  $\lfloor (i+j)/2 \rfloor$

    if x > a<sub>m</sub> then i:=m+1 else j:= m

**end**

if x=a<sub>i</sub> then location := i else location := 0

- How many comparisons are performed?
  - This will be studied in Section 2.3