

Discrete Mathematical Structures CS 2233 Lecture Twelve

Prof. William Winsborough
February 26, 2009

Business

- **Recall:**
 - Homework 6, due Thursday 3/5
 - 2.4: 2, 32, 36
 - 3.1: 2, 4, 12, 18, 24
 - Read Sections 2.4 and 3.1
- **Questions???**

Results from Midterm 1

If I had to assign grades today, they would look like this:

As	95	Bs	83	74	Cs	69	63	Ds	58	Fs	38
9	94	15	83	73	17	68	63	9	55	3	35
	92		83	72		68	62		54		29
	90		80	72		66	62		53		
	90		78	72		66	61		52		
	89		78	72		66	60		50		
	89		77	71		66	60		46		
	88		77			65	60		46		
	88					65	60		44		

Countability Exercise 1

- **Theorem:** If A is an uncountable set and B is a countable set, $A - B$ is uncountable
- **Proof**
 - Suppose for contradiction that $A - B$ is countable
 - This means that there is a sequence that enumerates all elements of $A - B$
 - We can now construct a sequence that enumerates A
 - It alternates between the sequence that enumerates $A - B$ and the sequence that enumerates B
 - This contradicts the assumption that A is uncountable
 - It follows that the assumption $A - B$ is countable is false, hence $A - B$ is uncountable

Countability Exercise 2

- **Theorem:** If A is an uncountable set and A is a subset of B , then B is uncountable
- **Proof:**
 - Suppose B is countable
 - This means there is a sequence that enumerates B
 - A sequence that enumerates A can be constructed by dropping the elements of $B - A$, yielding the desired contradiction

Algorithms

- **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 (Finiteness), Generality**

Examples

- Searching
 - Determine whether a given value is contained in an input sequence of integers and, if it is, returning its location
- Sorting
 - Taking an input sequence of (say) numbers, and reordering the sequence to be in order

26 February 2009

Winsborough CS 2233 Lecture 12

7

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?

26 February 2009

Winsborough CS 2233 Lecture 12

8