CS 113
DISCRETE STRUCTURES

Chapter 2: The Language of Mathematics

HOMEWORK

- Section 2.1: 1-10, 25-29, 31, 33, 35, 78, 82
- Section 2.2: 4, 6, 9, 43, 45, 46, 74, 88-92, 96
- Section 2.3: 1-31 odd only

SEQUENCES

- **A sequence is a list of things**
  - **For example**
    - **1, 3, 4, 5, 8, 9, -3, -5**
- **Our sequences will contain only numbers, though they can actually contain anything**
- **Sequences can be infinite or finite**
  - **Ours will be finite**
- **We will refer to a sequence as follows**
  - $a_1, a_2, a_3, a_4, \ldots, a_n$
- **Each item in a sequence is called a term**

IDEAS ABOUT SEQUENCES

- **If the elements in the sequence satisfy $a_i < a_j$ whenever $i < j$, the sequence is increasing**
  - **For example, the sequence 1, 3, 5, 7, 9, 11, … is increasing**
- **If the elements in the sequence satisfy $a_i > a_j$ whenever $i < j$, the sequence is decreasing**
  - **For example, the sequence 7, 5, 3, 1, … is decreasing**
- **Also, if you take some of the terms, it is called a subsequence**

COMBINING THE TERMS

- If we add the terms of a sequence, we call it a series
- We can also multiply the terms
- Then we get a product
- It may also be infinite or finite

NOTATION

- **We will write a series like this**

$$\sum_{i=1}^{n} a_i$$

- **This will mean $a_1 + a_2 + a_3 + \ldots + a_n$**

- **It's similar for a product**

- $\prod_{i=1}^{n} a_i$ **will mean $a_1 a_2 a_3 \ldots a_n$**

CHECKING ON THE NOTATION

- **Let's try #10, 22 on p. 69**

RECURSIVE DEFINITIONS

- **Sequences and series can also be defined recursively**
- **For example, p. 69, #44**

VOCABULARY

- In the series $\sum_{i=1}^{n} a_i$
- i is the index
- 1 is the lower limit
  - It doesn't have to be 1 of course
- n is the upper limit
- a_i is called the summand
- It's almost exactly the same for a product

SHIFTING THE INDEX

- **This is a technique**
  - **It's often used to make two sequences "match up"**
- **Suppose we have** $\sum_{i=1}^{n} a_i$
- **and we want to change the index to i-1**
- **It's very easy**
- **The index is i**
  - **We need a new variable, say j**
  - **If we want to change the index to i-1, we should let j = i – 1**
- **Solving for i gives i = j+1**
- **We just substitute j+1 for i—everywhere**

COMPLETING THE SHIFTING—AN EXAMPLE

- **Suppose the original sequence is $\sum_{i=1}^{5} i+3$**
  - **This means the sum is (1+3) + (2+3) + (3+ 3) + (4+3) + (5+3),**
  - **which is 4 + 5 + 6 + 7 + 8**
- **The lower limit is 1**
  - **This means i = 1. Substituting gives j+1 = 1, or j = 0.**
- **The upper limit is 5**
  - **Just like above. This means i = 5. Substituting and solving gives j = 4.**
- **The summand is i + 3. Changing to j gives (j+1) + 3**

COMPLETING THE EXAMPLE

- **The sum is now** $\sum_{j=0}^{4} j+4$
- **This is 4 + 5 + 6 + 7 + 8**
- **Comparing that to the original series, we see that they are the same**

NUMBER SYSTEMS

- Our number system is based on 10
- 8,142 is 8000 + 100 + 40 + 2
- This is $8 \bullet 10^3 + 1 \bullet 10^2 + 4 \bullet 10^1 + 2$
- Notice that our digits go from 0 – 9
    - 9 is 1 less than the base of our number system
- Computers live in a base 2 number system
    - That means a number can only have 0 and 1 for digits
    - Base 2 numbers are called binary numbers
- Remember, we are not inventing new math here
    - We are only renaming the numbers we already know

THE NEED FOR OTHER BASES

- It's too cumbersome to write numbers in binary
  - For example the base 2 number 10,000,000,000 interpreted as a binary number is 1024
  - This is not good!
- So, we group the digits in 4s and call it hexadecimal or hex for short
  - Hexadecimal means base 16
- This means the digits have to go from 1 to 15, which is 1 less than 16
- Our decimal numbers only have digits from 1 to 9
- We need new digits
- We use A for 10, B for 11, C for 12, D for 13, E for 14, and F for 15

Converting between Bases

- **It's not too hard to convert numbers between bases**
- **Binary to decimal**
- **Method 1:**
  - **Expand in powers of 2 and add**
- **Method 2:**
  - **Use nested multiplication**
- **Let's convert 1001_2 to base 10**

MORE CONVERSIONS

- **Converting binary to hexadecimal**
- **Memorize this table**
  - **It's not hard to figure out if you don't have it memorized**
- **Convert numbers in groups of four digits**
- **Let's convert 100111010011_2 to hex**

Hex	Binary	Hex	Binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

ONE MORE CONVERSION

- **Converting hex to binary is also not too hard**
- **You just use that table again**
- **Let's convert BEEF$_{16}$ to binary**

EVEN MORE CONVERSIONS

- To convert from decimal to binary
  - Divide by 2
  - Ignore the result but keep the remainder
- Repeat those steps until you get a quotient of 0
- Read the digits in reverse order
- Let's convert 43 into binary

- Converting to hex uses the same procedure, but this time you divide by 16
- You also have to remember the "unusual digits" in hex
  - For example, if the remainder is 10, the hex digit should be A

ADDITION

- **Binary addition is easy**
- **Initially, the usual rules apply**
  - **0+0 = 0**
  - **0+1=1**
    - **Addition is commutative again, so we also get the other rule: 1+0=1**
- **But, there is one unusual rule**
  - **1 + 1 = 10**
  - **That is, 1+1=0 and carry 1**
- **Wait! Is this a new rule?**
  - **No, 10_2 is really 2**
- **Let's try to add 10010011 and 01010101**

HEX ADDITION

- It's not too different
  - You just have to memorize a new addition table

OCTAL MATH

- **We can also do base 8 math**
  - **This is called octal math**
- **It's quite similar to what we have seem**
- **There was a very popular computer company in the late 1960s through the 1990s that used octal**
- **There are still some of those machines running code**

QUESTIONS

- Any questions?