Math Review

(with a little code thrown in)
Math Review: Modulo

• remainder after division
  3 mod 4 = 3
  7 mod 3 = 1
  N mod 2 = 0 (for N even)
  N mod 2 = ? (for N odd)

• Congruency  \( A \equiv B \pmod{N} \)
  if A and B have the same remainder mod N
Math Review: Modulo 2

- Congruent symbol $\equiv$
  Normal arithmetic rules apply, just like $=$.

- Examples: if $A \equiv B \ (\text{mod} \ N)$ then

\[
A + C \equiv B + C \ (\text{mod} \ N)
\]
\[
AC \equiv BC \ (\text{mod} \ N)
\]

might help to imagine an $=$
Math Review: Exponents

- Exponent formulas

\[ x^A x^B = x^{A+B} \]

\[ \frac{x^A}{x^B} = x^{A-B} \]

\[ (x^A)^B = x^{AB} \]

- Example: What is \(2^N + 2^N\)?
  
  \[\text{hint: } 2^N + 2^N = 2 \cdot (2^N) = 2^1 \cdot 2^N\]
Math Review: Logarithms

• Definition

\[ \log_x B = A \iff x^A = B \]

• i.e., logs are the anti-exponential!

• This is the definition.
Math Review: Logarithms 2

• Useful Formulas

\[ \log_A B = \frac{\log_C B}{\log_C A} \]

\[ \log AB = \log A + \log B \]
Math Review: Logarithms 3

• An important one for speed of algorithms!

\[ \log x < x \quad (\text{for } x > 0) \]

• So if an algorithm runs in \( \log(x) \) time, it runs faster than an algorithm that runs in \( x \) time.
Math Review: Logarithms 4

**Formula**

\[
\log(x^A) = A \log x
\]

**Proof:** Let .................

Take log of both sides ........

By definition \(\log_x B = A\) so ....

Recall \(\log_x B = \frac{\log B}{\log x}\) so ....

Hence .........................

\[
\begin{align*}
x^A &= B \\
\log_x x^A &= \log_x B \\
\log_x x^A &= A \\
\frac{\log(x^A)}{\log x} &= A \\
\log(x^A) &= A \log x
\end{align*}
\]
Math Review: Examples?

- What do the following equal?

\[ x^A + x^B = ? \]

\[(\log A) \cdot (\log B) = ? \]

\[ \log_{10} 10^x = ? \]
Math Review: Series 1

- Summation symbol – greek letter “S”

\[
\sum_{n=1}^{5} n = 1 + 2 + 3 + 4 + 5
\]

- Frequently sum to \( \infty \)

\[
\sum_{n=1}^{\infty} n = 1 + 2 + 3 + 4 + 5 + \ldots = \infty
\]
• Series are often represented by “for” loops in code.

\[ \sum_{n=1}^{5} n = 1 + 2 + 3 + 4 + 5 \]

```java
int total = 0;
for(int n=1; n<=5; n++)
{
    total += n;
}
```
Math Review: Series 3

• Another “for” loop series.

\[ \sum_{n=1}^{5} n^2 = 1 + 4 + 9 + 16 + 25 \]

int total = 0;
for(int n=1; n<=5; n++)
{
    total += n*n;
}

25 16 9 4 1
5 1 + + + +  
= n n
Math Review: Series 4

- Infinite series as a “while” loop.

\[ \sum_{n=1}^{\infty} n = 1 + 2 + 3 + 4 + 5 + \ldots = \infty \]

```java
int n = 0;
int total = 0;
while(true)
{
    total += n;
    n++;
}
```

Uh oh…
Math Review: Series 5

• Some infinite series converge…

\[ \sum_{i=0}^{\infty} A^i = \frac{1}{1-A} \]  

(for 0 < A < 1)

called “geometric” series

• Proof:

\[ S = 1 + A + A^2 + A^3 + ... \]
\[ AS = A + A^2 + A^3 + ... \]
\[ S - AS = 1 \]
\[ S = \frac{1}{1-A} \]
Math Review: Series 6

• Useful formulas:

\[ \sum_{i=0}^{N} A^i = \frac{A^{N+1} - 1}{A - 1} \quad \text{(for } A \text{ not 1)} \]

\[ \sum_{i=0}^{N} A^i \leq \frac{1}{1 - A} \quad \text{(for } 0 < A < 1) \]

\[ \sum_{i=1}^{N} i = \frac{N(N + 1)}{2} \approx \frac{N^2}{2} \]

called “arithmetic” series
Math Review: Proofs

- No big deal – already seen two!
- Types
  - proof by construction (e.g., series proofs)
  - proof by counterexample
  - proof by contradiction
  - proof by induction (often used in algorithm analysis)
  - proof by “Because I said so.”
    - Valid only for professors.
Proofs: Counterexample

- Best used to prove something is false.

- Example:

  **Theorem:** \( x^3 > x^2 \) for all \( x \).

  **Counterproof:** Let \( x = 1 \). Then \( 1 > 1 \). Obviously false. So theorem is false!
Proofs: Induction

- **Two steps**
  - **Base case:** Show that
    - Theorem holds for $n = c$
  - **Induction step:** Show that
    - If theorem holds for $n-1$ then theorem holds for $n$.

- **Why’s it work?**
  - If true for 1 (base case), then true for 2 (induction step)
  - If true for 2, then true for 3 (induction step)
  - If true for 3, then true for 4 (induction step)
  - etc. to infinity
Theorem: The nth odd number is $2n-1$.

Proof:
Base case. For $n=1$, $2(1)-1 = 1$, which is indeed the first odd number.

Induction step. S’pose true for $n-1$. Then the $(n-1)^{st}$ odd number is $2(n-1)-1$. Just have to add 2 to get the next odd number… So the nth odd number is $2(n-1)-1 + 2 = 2n-1$. Done!
Induction Example 2

Theorem: The sum of the first $n$ positive integers is $S(n) = n\,(n+1)/2$.

Proof:

Base case. For $n = 1$, the sum is just 1. And $1(1+1)/2 = 1$. So theorem is true for base case.

Induction step. Suppose $S(n-1) = (n-1)(n)/2$. Then

$$S(n) = S(n-1) + n$$

$$= \frac{(n-1)(n)}{2} + n$$

$$= \frac{n^2 - n + 2n}{2}$$

$$= \frac{n^2 + n}{2}$$

$$= n(n+1)/2 \quad \text{Aha!}$$
Induction Example 3

**Theorem:** The sum of the first \( n \) odd numbers is \( S(n) = n^2 \).

**Proof:**

**Base Case.** \( n=1 \) yields \( 1 = 1^2 \) which is true.

**Induction.** S’pose \( n-1 \) is true. Recall \( 2n-1 \) is the \( n \)th odd number! So 
\[
S(n) = S(n-1) + (2n-1)
\]
\[
= (n-1)^2 + (2n-1)
\]
\[
= n^2 - 2n + 1 + 2n - 1
\]
\[
= n^2
\]
Bingo!
Theorem: Every integer $n > 1$ is divisible by a prime.

Proof:

Base Case. $n=2$ is divisible by 2, which is prime.

Induction. Assume theorem is true for $n-1$ (and hence $n-2$, $n-3$, etc.).

Case 1: $n$ is prime. Then $n$ is divisible by itself.

Case 2: $n$ is not prime. In that case it must be true that $n = a \times b$ for some integers $a$ and $b$. And $a$ and $b$ are less than $n$. By induction $a$ and $b$ are divisible by a prime. Therefore $n$ is divisible by a prime.
Proofs: Contradiction

- Assume the theorem is false. Show this implies something contradictory and/or stupid (like $1=2$).

  - **Theorem:** There are an infinite number of primes.
  - **Proof:** Assume false. Then all the primes are $P_1, P_2, P_3, P_4, \ldots, P_k$ and $P_k$ is the biggest. Consider $N=1+(P_1P_2\ldots P_k)$. Clearly $N > P_k$ so it is not prime. But $N$ is not divisible by $P_1, P_2, P_3, P_4, \ldots$, or $P_k$ (remainder of 1). *Contradiction*, because every number is divisible by a prime (we just proved it by induction)! So assumption of false was wrong. In other words, theorem is true.
Math Review: Recursion

• Standard function
  • \( f(x) = x + 1 \)

• Recursive function
  • \( f(x) = f(x-1) + x + 1 \)
  • \( f(0) = 1 \)

So \( f(1) = f(0) + 1 + 1 = 1 + 1 + 1 = 3 \)
So \( f(2) = f(1) + 2 + 1 = 3 + 2 + 1 = 6 \)
So \( f(3) = f(2) + 3 + 1 = 6 + 3 + 1 = 10 \)

Defined in terms of itself!
Need both.
seed or “base case”

triangular numbers!
Recursion example 1

```java
public static int triangularNumber(int x)
{
    if(x==0)
        return 1;
    else
        return triangularNumber(x-1) + x + 1;
}
```

(Note: *chews up memory* with pending (unfinished) calls to the same function. e.g., call to f(1) can’t finish until call to f(0) is finished.)
Recursion example 2 (bad)

```java
public static int badFunction(int x) {
    if(x==0)
        return 0;
    else
        return badFunction(x/3+1) + x - 1;
}
```

(What can go wrong? Hint: Try x = 1. Circular…)
Recursion example 3

- Dictionary
  1. Look up pedantic.
  2. Means doctrinaire. (Huh?)
  3. Look up doctrinaire.
  4. Means inflexible. (Ahhhh…)

f(pedantic) = doctrinaire
f(doctrinaire) = inflexible
Recursion example 4 (bad)

- Dictionary
  (1) Look up **rigid**.
  (2) Which means **unbending**.
  (3) Which means **inflexible**.
  (4) Which means **rigid**.

Ouch! Circular. Never reaches seed.