Chapter 10-0 – Math Review & Run-Time Analysis

Exponents

Exponent Rules For $a \neq 0, b \neq 0$	
Product Rule	$a^x \times a^y = a^{x+y}$
Quotient Rule	$a^x \div a^y = a^{x-y}$
Power Rule	$\left(a^x\right)^y = a^{xy}$
Power of a Product Rule	$\left(ab\right)^x = a^x b^x$
Power of a Fraction Rule	$\left(\frac{a}{b}\right)^x = \frac{a^x}{b^x}$
Zero Exponent	$a^0 = 1$
Negative Exponent	$a^{-x} = \frac{1}{a^x}$
Fractional Exponent	$a^{\frac{x}{y}} = \sqrt[y]{a^x}$

Logarithmic Function

n=power (result obtained by raising b to the power of a)

$$\log_b n = a \text{ and } b^a = n$$

b=base

a=exponent

Logarithm Rules

$$\log_a xy = \log_a x + \log_a y$$

$$\log_a \frac{x}{y} = \log_a x - \log_a y$$

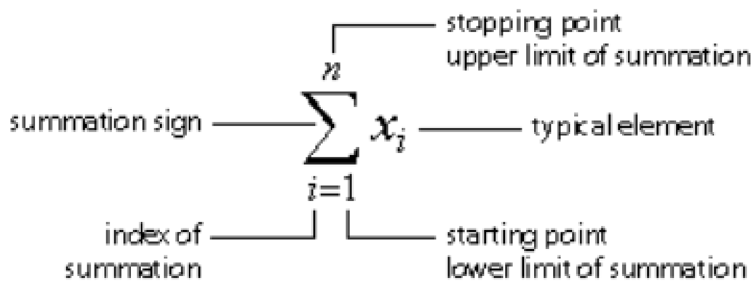
$$\log_a x^n = n \log_a x$$

$$\log_a b = \frac{\log_c b}{\log_c a}$$

$$\log_a b = \frac{1}{\log_b a}$$

Summation Notation

- Let $x_1, x_2, x_3, \ldots x_n$ denote a set of n numbers. x_1 is the first number in the set. x_i represents the ith number in the set.
- The summation sign Σ
  This appears as the symbol, Σ, which is the Greek upper case letter, S. The summation sign, Σ, instructs us to sum the elements of a sequence. A typical element of the sequence which is being summed appears to the right of the summation sign.



Some typical examples of summation

$\sum_{i=1}^{n} x_i$ This expression means sum the values of x, starting at x_1 and ending with x_n.

$$\sum_{i=1}^{n} x_i = x_1 + x_2 + x_3 + \cdots + x_n$$

$\sum_{i=1}^{10} x_i$ This expression means sum the values of x, starting at x_1 and ending with x_{10}.

$$\sum_{i=1}^{10} x_i = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}$$

$\sum_{i=3}^{10} x_i$ This expression means sum the values of x, starting at x_3 and ending with x_{10}.

$$\sum_{i=3}^{10} x_i = x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}$$

Summations used in this semester

1. $\sum_{i=1}^{n} i \ = 1 + 2 + 3 + \cdots + \ n = \frac{n(n+1)}{2}$
2. $\sum_{i=1}^{n} 1 \ = 1 + 1 + 1 + \cdots + \ 1 = \ n$
3. $\sum_{i=1}^{n} \ i^2 \ = 1^2 + 2^2 + 3^2 + \cdots + n^2 = \ \frac{n(n+1)\ (2n+1)}{6}$
4. $\sum_{i=0}^{n} \ a^i \ = a^0 + a^1 + a^2 + \cdots + a^n = \ \frac{a^{n+1}-1}{a-1}$ if a >1

   $\frac{1}{a-1} \quad if \ 0 < a < 1$

Run-Time Analysis

- An algorithm's execution time (run-time) is related to the number of operations it requires.
- Counting an algorithm's operations is a way to access its efficiency.
- The goal of run-time analysis is to obtain a "pen and paper" estimate of how efficient an algorithm or a program or a data structure is.
- More precisely, the running time of programs typically depends on the size of the input, and the goal of run-time analysis is to obtain a machine independent estimate of the running time of an algorithm or a program as a function of the size of the input.

Example 1) total running time? Count number of operations.

```
int i, sum=0;
for (i = 0; i < 3; i++)
       sum = sum + i;
```

```
int i, sum=0;

for (i = 0; i < 5; i++)

       sum = sum + i;
```

```
int i, sum=0;

for (i = 0; i < n; i++)

       sum = sum + i;
```

```
int i, sum=0, prod=1;

for (i = 1; i <= 3; i++)
{
   sum = sum + i;
   prod = prod * i;
}
```

```
int i, sum=0, prod=1;

for (i = 1; i <= n; i++)
{
   sum = sum + i;
   prod = prod * i;
}
```

```
int i,j, sum=0;

for (i = 1; i < 3; i++)
   for (j =1; j < 3; j++)
       sum = sum + i;
```

```
int i,j, sum=0;

for (i = 1; i < 5; i++)
   for (j =1; j < 5; j++)
       sum = sum + i;
```

```
int i,j, sum=0;

for (i = 1; i < n; i++)
   for (j =1; j < n; j++)
       sum = sum + i;
```