starting at page 115

Seven Functions to Analyze Algorithms

→ constant function: $f(n) = c$
- c is fixed constant (5, 27, or 2^{10}, etc.)

→ logarithm function: $x = \log_b n$ if $b^x = n$
- b is known as the base of the log

→ linear function: $f(n) = n$
- given input value n, function f assigns value n itself

→ n-log-n function: $f(n) = n \log n$
- function that assigns an input n the value of n times the log of n
- grows more rapidly than linear but less rapidly than quadratic

→ quadratic function: $f(n) = n^2$
- function assigns product of n squared
- often appears in analysis of nested loops where the inner loop performs a linear operation and outer loop is performed a linear amt. of times

→ cubic function: $f(n) = n^3$

Polynomials

→ function has the form: $f(n) = a_0 + a_1 n + a_2 n^2 + a_3 n^3 + \ldots + a_d n^d$
- a_0, a_1, etc. are constants or coefficients of the polynomial
- integer d is the degree (highest power/coefficient)

Summation: $\sum_{i=a}^{b} f(i) = f(a) + f(a+1) + f(a+2) + \ldots + f(b)$
where a & b are ints and $a \leq b$.

→ often arise in algorithm analysis b/c running times of loops

Exponential Function: $f(n) = b^n$

→ b is a positive constant (base) and argument n is exponent

Geometric Sums: suppose a loop where each iter. takes a factor longer than previous

→ for any $n \geq 0$ and any $a > 0$ and $a \neq 1$, consider: $\sum_{i=0}^{n} a^i = 1 + a + a^2 + \ldots + a^n$

→ called geometric summation b/c each term is larger than the previous one