1 How to Use

This reference includes formulas useful for analyzing the time or space complexity of algorithms. If the algorithm is recursive, the runtime boils down to calculating the time in each level of recursion, then summing all the levels. If the algorithm is iterative, it boils down to summing the time in each iteration. These summations can be complicated, hence the formulas in this reference sheet.

A useful fact: If a variable x is divided by a constant y at each step, it will reach 1 after about $\log_y x$ steps.

These formulas can be used flexibly. The last index n can be substituted for a function of n. For example:

$1^2 + 2^2 + 3^2 + \cdots + \log^2 n = \Theta(\log^3 n)$

$1 + 2 + 2^2 + \cdots + n = 1 + 2 + 2^2 + \cdots + 2^{\log n} = \Theta(2^{\log n}) = \Theta(n)$

2 Summation Formulas

Expanded Form	Summation	Exact Value	Complexity
$1 + 1 + 1 + \cdots + 1$	$\sum_{i=1}^{n} 1$	n	$\Theta(n)$
$1 + 2 + 3 + \cdots + n$	$\sum_{i=1}^{n} i$	$\frac{1}{2}n(1+n)$	$\Theta(n^2)$
$1^p + 2^p + 3^p + \cdots + n^p$	$\sum_{i=1}^{n} i^p$	N/A	$\Theta(n^{p+1})$
$1 + a + a^2 + \cdots + a^n$	$\sum_{i=0}^{n} a^i, a > 1$	$\frac{a^{n+1}-1}{a-1}$	$\Theta(a^n)$
$1 + a + a^2 + \cdots + a^n$	$\sum_{i=0}^{n} a^i, a < 1$	$\frac{1-a^{n+1}}{1-a}$	$\Theta(1)$
$\log 1 + \log 2 + \log 3 + \cdots + \log n$	$\sum_{i=1}^{n} \log i$	$\log(n!)$	$\Theta(n \log n)$