**Summations**

**Triangular Numbers**: (proof by induction)



**Example 1**: Determine the value of the sum 1 + 2 + 3 + … + 50

**Application**: Determine the exact number of statements executed in the following algorithm:

**stockProfit**

n = p.last

buy = 1

sell = 2

profit = p[2] – p[1]

for i = 1 to n-1

for j = i+1 to n

if p[j] – p[i] > profit

buy = i

sell = j

profit = p[j] – p[i]

println(“Buy on day ” + buy)

println(“Sell on day ” + sell)

println(“Profit: $” + profit)

**Finite Geometric Sum:**  as long as *r* ≠ 1



**Example 2**: Determine the value of the sum 3 + 6 + 12 + 24 + … + 3072

**Applications of geometric sums and logarithms to binary trees**

* A **rooted** binary tree has a root node and every node has at most two children.
* A node with no children is called a **leaf**.
* The **depth** (or level) of a node is the number of edges from the root to the node (i.e., it is the length of a shortest path from the root to the node).
* The **height** of a node is the number of edges on the longest path between that node and a leaf (the leaf must be an ancestor of the node).
* The **height** of a binary tree is the height of the root.
* A **perfect** binary tree is a binary tree in which all interior nodes have two children *and* all leaves have the same *depth* or same *level*.
* A **complete** binary tree is a binary tree in which every level, *except possibly the last*, is completely filled, and all nodes in the last level are as far left as possible.
* NOTE: A heap is a data structure that can be viewed as a complete binary tree with some added constraints on the order of the data in the tree. We will study heaps later.

**Example 3**: Find the number of nodes in a **perfect** binary tree of height 12.

**Example 4**: Find the range (min and max) for the number of nodes in a **complete** binary tree of height 6.

**Example 5**: Find an expression for the number of nodes in a perfect binary tree of height *h*.

**Example 6**: Find an expression for the height of a perfect (or complete) binary tree with *n* nodes (as a function of *n*).

**Example 7**: Find the height of a complete binary tree with 2000 nodes.

**Arithmetico-geometric series**: for *n* ≥ 1, 0 < *r* < 1



**Example 8**: Find an upper bound for the sum

