

# Sums of Powers Of Two

## Introduction

My goal is to explore the following relation:

$$1 + 2 + 4 + 8 + \dots + 2^n = 2^{n+1} - 1$$

I'll start by providing the standard proof by induction. Then I'll discuss how I personally reason about it and find it plausible. This article will include tree visualizations, focus on the recursive nature of this sum, discuss how I use approximations to simplify my reasoning, and finally, I'll conclude with 2 bonus sections where I walk through another sum of terms recurrence relation and where I discuss both problems from a Computer Science and programming perspective.

## Prerequisites

Knowledge of proof by induction. Which I believe people learn in high school when they take Precalculus or in their second Algebra course.

For the bonus CS section at the end, the main prerequisite is knowing binary arithmetic and generally having some introductory level experience.

## Proof By Induction

I'll focus on the left-hand side (LHS) and right-hand side (RHS) separately and express them as functions.

For the LHS, I'll define the function:

$$S(n) = 2^0 + 2^1 + \dots + 2^n$$

This function has  $n + 1$  terms. And here it is expressed as a sum:

$$S(n) = \sum_{i=0}^n 2^i$$

For the RHS, I'll define the function:

$$F(n) = 2^{n+1} - 1$$

Let  $P_n$  be the claim that  $S(n) = F(n)$ . I want to prove  $P_n$  is true for all  $n = 0, 1, 2, \dots$ . To this end, I'll use induction. The first step is to check the base case,  $P_0$ .  $S(0) = 2^0 = 1$  and  $F(0) = 2^{0+1} - 1 = 1$ .  $S(0) = F(0)$ . Done.

Next, if I can show  $P_k \Rightarrow P_{k+1}$ , I'll be done. Let's prove this the textbook way and then repeat it but with a focus on the recursive natures of  $S(n)$  and  $F(n)$ .

### Approach 1: Textbook Induction

So we do this

### Approach 2: Induction emphasizing recursive definitions

So we do this