What is a polynomial?

A polynomial is an algebraic expression consisting of variables and coefficients combined using addition, subtraction, and multiplication, where the exponent of the variable is a non-negative integer. A general polynomial in one variable can be written as:

$P(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$

Here, $a_0, a_1, a_2, \ldots, a_n$ are coefficients and the highest power of x determines the degree of the polynomial.

In programming, a polynomial can be efficiently represented using a one-dimensional array. In this representation, the **index of the array corresponds to the power of x**, and the value stored at that index represents the coefficient of that term. For example, the polynomial:

$P(x) = 4x^3 + 3x + 7$

can be represented as an array:

P[0] = 7
P[1] = 3
P[2] = 0
P[3] = 4

This representation is especially effective for dense polynomials where most degrees are present.

Polynomial addition involves combining like terms, meaning that coefficients of the same power of x are added together. When two polynomials are represented as arrays, polynomial addition becomes equivalent to adding corresponding elements of the two arrays. If the polynomials have different degrees, the resulting polynomial will have a degree equal to the maximum degree among the two.

Why use arrays for polynomials?

In programming, we need a structured way to store:
- Coefficients
- Corresponding powers of x

Arrays are suitable because:
- Polynomial terms have a natural order (by degree)
- Coefficients can be accessed using indices
- Addition becomes simple when degrees are aligned

Array representation of a polynomial

Method: Index = Power of x

For the polynomial:
$P(x) = 5x^3 + 4x^2 + 2x + 7$

We use an array P such that:
$P[0] = 7$
$P[1] = 2$
$P[2] = 4$
$P[3] = 5$

General rule:
$P[i] = \text{coefficient of } x^i$

This representation assumes:
• Polynomial is single-variable
• Exponents are non-negative integers

Example of two polynomials

$P_1(x) = 3x^3 + 2x + 5$
$P_2(x) = 4x^3 + 6x^2 + 1$

Array representation:

$P_1 \rightarrow [5, 2, 0, 3]$
$P_2 \rightarrow [1, 0, 6, 4]$

(Index 0 → x^0, Index 1 → x^1, etc.)

Concept of polynomial addition

Polynomial addition follows one simple rule:

Add coefficients of like powers

Mathematically:
$(P_1 + P_2)(x) = \Sigma\ (P_1[i] + P_2[i]) \cdot x^i$

This maps directly to array addition.

Algorithm for polynomial addition using arrays

1. Determine the maximum degree of both polynomials

2. Create a result array of size (maxDegree + 1)

3. For each index `i`:
   • result[i] = $P_1[i] + P_2[i]$

4. The result array represents the sum polynomial

Pseudocode

```
Input: Array P1, Array P2
Output: Array Sum

maxDegree = max(length(P1), length(P2))
create array Sum of size maxDegree

for i = 0 to maxDegree - 1
    if i < length(P1) and i < length(P2)
        Sum[i] = P1[i] + P2[i]
    else if i < length(P1)
        Sum[i] = P1[i]
    else
        Sum[i] = P2[i]
end for
```

Time and space complexity

Time Complexity:
• O(n), where n is the maximum degree

Space Complexity:
• O(n) for the result array

This is optimal for dense polynomials.

Advantages of array-based representation

• Simple and intuitive
• Efficient for dense polynomials
• Direct index access
• Easy to implement addition and subtraction

Limitations

• Wastes space for sparse polynomials

- Not suitable when degrees are very large but terms are few
- Linked lists are better for sparse cases