

# Power Function ( $x^n$ ) using Binary Exponentiation

## Concept:

The problem is to implement  $\text{pow}(x, n)$ , which calculates  $x$  raised to the power  $n$  ( $x^n$ ).

## Approach:

We use **Binary Exponentiation** (also known as Exponentiation by Squaring).

- If  $n$  is negative, we convert the problem to  $1/x$  and make exponent positive.
- Repeatedly square the base and reduce exponent by half.
- If exponent is odd, multiply result by base.

**Time Complexity:**  $O(\log n)$

**Space Complexity:**  $O(1)$

```
class Solution {
public:
    double myPow(double x, int n) {
        double ans = 1;
        long long exp = n;
        if(n < 0){
            x = 1 / x;
            exp = -exp;
        }

        while(exp != 0){
            if(exp % 2 == 1){
                ans *= x;
                exp--;
            }

            else{
                x *= x;
                exp /= 2;
            }
        }

        return ans;
    }
};
```