pow(x, n)

$x * x * x * \cdots * x$ (n times)

Brute Force :-

We will multiply x, n times.

$T = O(n)$

$S = O(1)$

Best Approach :-

Example 1 :

$7^9 = 7 \times 7^8$

$7^8 = (7 \times 7)^4 = (49)^4$

$(49)^4 = (49 \times 49)^2 = (2401)^2$

$(2401)^2 = (2401 \times 2401)^1 = (5764801)^1$

$(5764801)^1 = 5764801 \times (5764801)^0$ (power 1 → 0)

Example 2 :

$2^{10} = (2 \times 2)^5 = 4^5$

$4^5 = 4 \times 4^4$

$4^4 = (4 \times 4)^2 = 16^2$

$16^2 = (16 \times 16)^1 = 256^1$

$256^1 = 256 \times (256)^0$

$(256)^0 = 1$

```
public int power(int x, int n) {
    if (n == 0) {
        return 1;
    } else if (n % 2 == 0) {
        return power(x * x, n/2);
    } else {
        return x * power(x, n-1)
    }
}
```

And also remember $x^{-n} = \frac{1}{x^n}$. This case also has to be handled .i.e, when $n < 0$.

Whenever n is even we reduce it by dividing it by 2. And whenever n is odd, we reduce n by subtracting 1 from it, then n becomes even and then again we reduce n by dividing it by 2.

$$T = O(\log_2(n))$$