

50. Pow(x, n)

Total Accepted: 75047

Total Submissions: 273134

Difficulty: Medium

Implement $\text{pow}(x, n)$.

Analysis:

Try to use recursion.

Case 1: if n is even, for example, $n = 16$.

$\text{Pow}(x, 16) = \text{Pow}(x^2, 8) = \text{Pow}(x^4, 4) = \text{Pow}(x^8, 2) = \text{Pow}(x^{16}, 1)$

$\text{Pow}(x, 16) = \text{Pow}(x^2, 16/2) = \text{Pow}((x^2)^2, 16/2/2) = \text{Pow}(((x^2)^2)^2, 16/2/2/2) = \text{Pow}((((x^2)^2)^2)^2, 16/2/2/2/2)$

Case2: What if n is odd number, for example, $n = 17$. That's easy too.

$\text{Pow}(x, 17) = x * \text{Pow}(x, 16)$

In other word, if n is odd, we set $\text{Pow}(x, n) = x * \text{Pow}(x, n-1)$, now that the $n-1$ in $\text{Pow}(x, n-1)$ is even number, we can use the the same method to solve it, as in case 1.

Case3: If $n < 0$, for example, $n = -16$. Because $\text{Pow}(x, -16) = 1 / \text{Pow}(x, 16)$

So, if $n < 0$, we can set $\text{Pow}(x, n) = 1 / \text{Pow}(x, -n)$

Code: Complexity is $O(\log n)$

```
3 class Solution(object):
4
5     def myPow(self, x, n):
6
7         if n == 0:
8             return 1.0
9
10        elif n == 1:
11            return x
12
13        elif n < 0:
14            return 1.0 / self.myPow(x, -n)
15
16        else:
17            if n % 2 == 0:
18                return self.myPow(x*x, n/2)
19            else:
20                return self.myPow(x*x, (n-1)/2) * x
21
```