Exercise 2: Calculate 2^n

1. Algorithm 1: Use the defined linear search : $2^n = 2^{n-1} * 2$
   a. Pseudo-code:

```
result = 1
for i from 1 to n:
        result = result * 2
```

   b. Complexity: $O(n)$
2. Algorithm 2: Use the divide and conquer:

$$2^n = (2^{n/2} * 2^{n/2}) \; if \; (n \,\%\, 2 == 0)$$
$$2^n = (2^{n/2} * 2^{n/2} * 2) \; if \; (n \,\%\, 2 == 1)$$

   a. Pseudo-code:

```
function pow(a, b):
        if b == 0 return 1
        res = pow(a, b / 2)
        res = res * res
        if b % 2 == 1:
        res = res * a
        return res
pow(2, n)
```

   b. Complexity: $O(logn)$