## Question 1

To compute , we can recursively compute . If n is odd, . If n is even, . The time complexity could be reduced by repeating square.

For example,

In such way, we compute and using only three multiplication. For , it has at most times multiplications.

Pseudo code:

1. **if** M == 0:
2. **return** 0.0
3. num = 1
5. **while** n:
6. **if** n & 1:
7. res \*= M    // when n is odd
8. M \*= M
9. n >>= 1   // n/2
10. **return** res

This method is the implementation of divide and conquer. It is using times multiplications.