Q1:

Given： positive integers M and n

Output： a methods with O(logn) multiplications

Solution:

According to exponential product rules, , then we can rewrite our n in binary as n = , then is equivalent to , repeat this process until n = 1 and stop. The Pseudocode shown as

Input: integer m and n;

Output: the n power of m

Function Rescurssion(m,n):

If n=0 then

Return 1;

End if

If n%2=0 then

Return square(square(m,n/2))

End if

Else

Return m\*square(square(m,n/2))

End if

End function

The height of this Recurrance tree with in the range 0 to , So that, Totally there will be at most 2\* multiplications. The square multiplication can be solved by bits shift in one time each. Then there will have O(logn) times multiplications.