Mubarra Khan z5161249 Tuesday, 23 June 2020

# Q1

---

Given positive integers *M* and *n* compute $M^n$ using only $O(logn)$ many multiplications.

<u>Answer:</u>

This could be achieved using a recursive function which firstly writes n in binary i.e., as $n = 2^{k1} + 2^{k2} + 2^{km}$ where $k_1 > k_2 > \ldots k_m$ . The function creates these bits by a recursive call where n = n/2 at each call and storing $M^n$ in a temporary variable which is at most $log_2 n$ function calls.

At each call *if* n is even -> then the temp variable is **squared** $M^{2^j} = M^{2^i} * M^{2^i}$

since n is of the form $n = 2^{k1} + 2^{k2} + 2^{km}$

Else if n is odd $M^{2^j} = M * M^{2^i} * M^{2^i}$

That is enough to compute all of $M^{2^j}$ where $1 \leq i < j \leq log_2 n$ with at most $log_2 n$ multiplications