z5242692

Chenqu Zhao

COMP9101 (T2-2020)

Homework 2 - Q1

First, we write n in binary format, $n = 2^{k_1} + 2^{k_2} + \cdots + 2^{k_m}$ where $k_1 > k_2 > \cdots > k_m$ and $K_1 = \lfloor \log_2 n \rfloor$. This way $M^n = M^{2^{k_1}} \cdot M^{2^{k_2}} \cdots M^{2^{k_m}}$, which takes $\lfloor \log_2 n \rfloor - 1$ multiplications.

To compute $M^{2^j} (1 \leq j \leq \lfloor \log_2 n \rfloor)$, we apply repeated squaring algorithm as follows. As the value $k_1, k_2, k_3 \cdots k_m$ is continuous integer, we can compute $M^{2^{k_m}}$ first, then multiply it by $M^{2^{k_m}}$ each time to obtain other values of M^{2^j}. There are 2^j values which requires $2^j - 1$ multiplications. Since that $2^j \leq 2^{k_1} \leq 2^{\log_2 n} = n$, the computation of all M^{2^j} take at most $\lfloor \log_2 n \rfloor - 1$ multiplications.

Therefore, the total multiplications time is $\lfloor \log_2 n \rfloor + \lfloor \log_2 n \rfloor - 2 = O(\log n)$.