Dheeraj Satya Sushant Viswanadham
(z5204820)

COMP3121 Assignment 2 – Question 1

1) In this question, we have two positive integers – *M* and *n* and we are required to find $\mathbf{M^n}$ using only **O(log(n))** many multiplications.

We can employ the use of binary to help simplify our method, specifically writing our *n* in binary i.e.

$$n = 2^{k_1} + 2^{k_2} + \cdots + 2^{k_m}$$

where

$$k_1 > k_2 > \cdots > k_m$$

and

$$k_1 = \lfloor log_2(n) \rfloor$$

Once we have that, we can then rewrite M^n as $M^n = M^{2^{k_1}} * M^{2^{k_2}} * \ldots * M^{2^{k_m}}$. This essentially means that we work out M^{2^k} for all *k* where $2^k \leq n$. This can be seen to involve at most $\lfloor log_2(n) \rfloor$ multiplications. By computing all of m^{2^j} for all $1 \leq j \leq \lfloor \log_2(n) \rfloor$ this will require at most $\lfloor log_2(n) \rfloor$ multiplications. This can be done by repeated squaring where the number of digits within n is proportional to $\log_2(n)$ multiplications at most. Put more simply, as an example if we had 5^{17}, we can convert 17 into binary which would be 10001, then by reading from right to left (or from the least-significant bit to most-significant bit) we would find that it would be:

$$\begin{aligned} 5^{17} &= 5^{10001} \\ &= 5^{(1*2^4+0*2^3+0*2^2+0*2^1+1*2^0)} \\ &= 5^{1*2^4} * 5^{0*2^3} * 5^{0*2^2} * 5^{0*2^1} * 5^{1*2^0} \\ &= 5^{2^4} * 1 * 1 * 1 * 5^{2^0} \\ &= 5^{16} * 5^1 \end{aligned}$$

These final values to be multiplied are referred to as the successive squares (16 and 1) and we needed at most 4 "squarings" (as we know that the left-most digit is in the 2^4 position and the right-most number is just 5 as $5^{2^0} = 5^1 = 5$ hence we do not need to include that as a "squaring"). Intuitively, from above we recognise that the number of multiplications required would be at most *m* bits required to represent our exponent and hence would be a maximum of m – 1 multiplications. Putting it all together, the number of bits required to represent our positive integer *n* would be: $1 + \lfloor \log_2(n) \rfloor$. Note, however, that in most cases it is simply enough to find the number of bits from doing $\lfloor \log_2(n) \rfloor$ and then computing the required multiplications from there. In summation, this would lead to an overall maximum of $O(\log(n))$ multiplications, as required.

End of Solution