EXERCISE FOR CSE202 – WEEK 1

Exercise 1. *The lecture showed that the number of multiplications needed to compute an nth power is lower bounded by $\lfloor \log_2 n \rfloor$, while the binary powering algorithm needs at most twice as many multiplications. This exercise studies a variant of the binary powering algorithm that is asymptotically optimal, ie, does not have this extra factor 2 in its complexity.*

First, consider the following algorithm:

1. *Compute $1, x, x^2, x^3$;*
2. *Compute recursively x^n as $x^{n \bmod 4} \times (x^{n \operatorname{div} 4})^4$.*

(1) *Show that the number of multiplications required to compute x^n by this algorithm is at most*
$$3 \left\lfloor \frac{\log_2 n}{2} \right\rfloor + 2.$$
(2) *Propose a generalization of this algorithm, where 4 is replaced by $m = 2^k$ for a positive integer k, adjusting the first step as necessary. (For $k = 1$, you should recover binary powering.)*
(3) *Show that the number of multiplications required to compute x^n by this generalized algorithm is upper bounded by*
$$\log_2 n \left(1 + \frac{1}{k} + \frac{2^k}{\log_2 n} \right).$$
(4) *Show that the choice*
$$k = \lfloor \log_2 \log_2 n - \log_2 \log_2 \log_2 n \rfloor$$
leads to an asymptotically optimal algorithm.
(5) *This algorithm is mostly of theoretical interest for $k > 2$. Can you see why?*