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?*

Solution : 1. First, two multiplications are needed to compute x^2 and x^3. Next, the number of multiplications needed to compute x^n satisfies

$$C(n) \leq C(n \operatorname{div} 4) + 3,$$

since given $x^{n \operatorname{div} 4}$, its fourth power is obtained by two squarings and one more multiplication is needed to multiply by $x^{n \bmod 4}$ when it is different from 1.

The number of recursion steps is bounded by the number of times n can be divided by 4 before becoming smaller than 4, which is $\lfloor \log_4 n \rfloor = \lfloor \log_2 n / \log_2 4 \rfloor$, whence the answer.

2. The algorithm becomes:

(1) Compute $1, x, \dots, x^{m-1}$;
(2) Compute recursively $x^n = x^{n \bmod m} \times (x^{n \operatorname{div} m})^m$.

3. The analysis follows exactly the same steps as in question 1: first the number of multiplications is seen to satisfy

$$C(n) \leq C(n \operatorname{div} m) + k + 1.$$

Next estimating the number of recursion steps leads to the bound

$$(k+1)\left\lfloor \frac{\log_2 n}{k} \right\rfloor + m - 2.$$

The conclusion follows from $\lfloor \log_2 n/k \rfloor \leq \log_2 n/k$ and $2^k - 2 \leq 2^k$.

4. For the given choice of k, $2^k \leq \log_2 n/\log_2\log_2 n$, so that both the terms $1/k$ and $2^k/\log_2 n$ tend to 0 as $n \to \infty$ and the upper bound is equivalent to $\log_2 n$.

5. The growth of k with n is extremely slow: even if one takes the closest integer to this value of k, the smallest value of n when it becomes larger than 2 is 121,233,869, larger than 10^9, which is a huge exponent. $\square$