CSE202 - DESIGN AND ANALYSIS OF ALGORITHMS

Week 1 - Homework 1

September 26, 2022

Yubo Cai

EXERCISE 1

1.1

First, we want to compute x^2 and x^3. We have $x^2 = x \cdot x$ and $x^3 = x^2 \cdot x$ which need 2 multiplications for computation. Then, since we compute recursively x^n in the algorithms $x^{n \bmod 4} \times \left(x^{n \operatorname{div} 4}\right)^4$, we have the inequality of $C(n)$ that

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

Since we have $x^{n \operatorname{div} 4}$ and we compute it in the fourth power, therefore we need to compute $(x^{n \operatorname{div} 4})^2$ and then square it again in order to get $(x^{n \operatorname{div} 4})^4$, which needs 2 multiplications. Also we have to multiple $(x^{n \operatorname{div} 4})^4$ and $x^{n \bmod 4}$ which requires another multiplication. For this reason, we get 3.

The number of recursion steps is bounded by the number of times n and we divide by 4 each time. Therefore, we have the height of the tree which is $\lfloor \log_4 n \rfloor = \lfloor \log_2 n / \log_2 4 \rfloor = \left\lfloor \frac{log_2 n}{2} \right\rfloor$ (This can simply prove by computing $4^k = n$ where k in the height of the tree).

For each divide, we require 3 times of multiplications and when n becomes less than 4, we need 2 times to compute x^3. Then we prove that

$$C(n) \leq 3 \left\lfloor \frac{\log_2 n}{2} \right\rfloor + 2$$

1.2

The algorithms are similar to above:

1. Compute $1, x, x^2, x^3$ to x^{m-1};
2. Compute recursively x^n as $x^{n \bmod m} \times \left(x^{n \operatorname{div} m}\right)^m$

1.3

We apply the same method as in Exercise 1.1 to compute the number of multiplications required to compute x^n

We first have the inequality of $C(n)$ that

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

Since we have $x^{n \operatorname{div} m}$ and we compute it in 2^k power, therefore we need to compute it for k times and multiple $(x^{n \operatorname{div} m})^m$ and $x^{n \bmod m}$ which requires another multiplication. For this reason, we get $k+1$. We estimate the number of recursion steps we have

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

where $m-2$ is the number of multiplications needed to compute $1, x, ..., x^{m-1}$. We can deduce that

$$\begin{aligned}
(k+1)\lfloor \log_m n \rfloor + m - 2 &= (k+1)\left\lfloor \frac{\log_2 n}{k} \right\rfloor + m - 2 \\
&\leq (k+1)\frac{\log_2 n}{k} + 2^k \\
&\leq (1+\frac{1}{k})\log_2 n + 2^k \\
&\leq \log_2 n \left(1 + \frac{1}{k} + \frac{2^k}{\log_2 n}\right)
\end{aligned}$$

Then we finish the prove.

1.4

For the given choice of $k = \lfloor \log_2 \log_2 n - \log_2 \log_2 \log_2 n \rfloor$, we have $2^k \leq \frac{log_2 n}{log_2 log_2 n}$. Therefore, we have $\frac{1}{k}$ tend to 0 as $n \to \infty$ since k is divergent. Also, $\frac{2^k}{\log_2 n} \leq \frac{1}{log_2 log_2 n}$ and $\frac{1}{log_2 log_2 n}$ converge to 0 as $n \to \infty$. So we prove that the upper bound is equivalent to $log_2 n$.

1.5

Since we have the optimal algorithm of $k = \lfloor \log_2 \log_2 n - \log_2 \log_2 \log_2 n \rfloor$ and k grows really slow with the increase of n. And we need at least $n \geq 10^9$ so that we can have $k > 2$ which is a really large power. Therefore we need this large exponent so that we can see the improvement of optimal algorithm.