John Levy

- CSE202 -

Homework 1

For 13/09/21

1) First of all, we can easily notice that the number of recursion steps is limited by the number of times we can divide n (n being an integer) by 4 before it becomes smaller than 4.
To find this number let us assume that we try to divide n by 4^k, with $k \in \mathbb{N}$.
This gives, $1 \leq n/4^k < 4$

$$\Leftrightarrow 1 \leq n/2^{2k} < 2^2 \text{, we take the log in base 2 on both sides}$$
$$\Leftrightarrow \log_2(1) \leq \log_2(n/2^{2k}) < \log_2(2^2)$$
$$\Leftrightarrow 0 \leq \log_2(n) + \log_2(2^{-2k}) < 2$$
$$\Leftrightarrow 0 \leq \log_2(n) - 2k < 2 \Leftrightarrow 2k \leq \log_2(n) < 2 + 2k$$
$$\Leftrightarrow k \leq \frac{\log_2(n)}{2} < 1 + k$$

We recognise the definition of the floor function,
thus $k = \left\lfloor \frac{\log_2(n)}{2} \right\rfloor$

Secondly, at every recursion there is a maximum number of multiplications that can be done. When putting $x^{n \operatorname{div} 4}$ to its fourth power we do two squarings, and we can do one other multiplication when $x^{n \bmod 4}$ is different than 1. (So n is not a multiple of 4).

Finally, we do 2 more multiplications in the base case, when computing x^2 and x^3.
This gives that the number of multiplications to compute x^n with this algorithm is at most $3 \cdot \left\lfloor \frac{\log_2(n)}{2} \right\rfloor + 2$.

We can check that this satisfies the more formal complexity analysis:
$C(n) \leq C(n \operatorname{div} 4) + 3$ (the complexity is bounded by the ~~complexity~~ of number of times we can divide n by 4 and we add the three multiplications like discussed in the 2nd paragraph.

2) Let $m = 2^k$, $k \in \mathbb{N}$.

The algorithm is now:

1. Compute x^i for i ranging from 0 to $m-1$ (So $1, x, x^2, \dots, x^{m-1}$)
2. Recursively compute x^n as $x^{n \bmod m} \times (x^{n \operatorname{div} m})^m$.

3) We do an analysis similar to the one discussed in question 1.
The number of multiplication needs to satisfy the following inequality:

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

The number of recursion steps can be found using the same reasoning and gives: $(k+1)\left\lfloor \frac{\log_2(n)}{k} \right\rfloor + m - 2$.

Moreover, $\left\lfloor \frac{\log_2(n)}{k} \right\rfloor \leq \frac{\log_2(n)}{k}$ and $2^k - 2 \leq 2^k$.

which gives the desired result.

4) Let $k = \lfloor \log_2 \log_2(n) - \log_2 \log_2 \log_2(n) \rfloor$, clearly $2^k < \frac{\log_2(n)}{\log_2 \log_2(n)}$.

This gives that $\frac{1}{k}$ and $\frac{2^k}{\log_2(n)}$ tend to 0 as n tends to infinity.

Thus the upper bound is essentially $\log_2(n)$ and the algorithm is asymptotically optimal.

5) It is clear that k grows extremely slowly compared to n. As soon as we have $k > 2$, we can use this algorithm for extremely large values of n which makes it mostly of theoretical interest.