Notes class 08/28

I- Tower of Hanoi

$$T(n) = 2T(n-1) + 1$$

$$\Rightarrow T(n) = 2(\underbrace{2T(n-2)+1}_{\text{1st substitution}}) = 2^2 T(n-2) + 2^1 + 2^0$$

$$T(n) = 2^2(\underbrace{2T(n-2)+1}_{\text{2nd substitution}}) + 2^1 + 2^0 = 2^3 T(n-3) + 2^2 + 2^1 + 2^0$$

We substitued T two times here. We see a pattern and can guess that if we substitute k times we have:

$$T(n) = 2^{k+1} T(n-(k+1)) + 2^k + \ldots + 2^0$$

How many times should we substitute until we reach the basic case T(1) ? We should do it such that:

$n-(k+1) = 1 \iff \boxed{k = n-2}$, and if we substitute $k=$ n-2 times should we have:

$$T(n) = 2^{n-1} T(1) + \ldots + 2^0 \quad \text{thus } T(n) = \sum_{i=0}^{n-1} 2^i = \frac{1-2^n}{1-2} = \boxed{2^n - 1}$$

To make the proof mathematically sound you should prove it by recursion. Before is to give intuition of T(n).

Here is an example of proof by recursion:

- If $n=1$, then $T(n) = 2^1 - 1 = 1$, the formule is true for $n=1$

- Suppose the formule is true for n, is it true for $n+1$?

$$T(n+1) = 2T(n)+1 \underset{\text{by hypothesis } T(n)=2^n-1}{\longrightarrow} 2(2^n-1)+1 = 2^{n+1}-1$$ thus the formula is true for $n+1$

- By the principle of recursion, since the basic and the recursion step are true, it is true for all integers n.

For the exam, if you show you understood well using your intuition (if you explain well your results), it is sufficient. A proof by recursion is, however, better.

II- Binary search

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

1st subs.

$$T(n) = \left(T\left(\frac{n}{2^2}\right)+1\right)+1 = T\left(\frac{n}{2^2}\right)+1+1$$

2nd subs.

$$T(n) = \left(T\left(\frac{n}{2^3}\right)+1\right)+1+1 = T\left(\frac{n}{2^3}\right)+1+1+1$$

So if we substitute k times we should have $T(n) = T\left(\frac{n}{2^{k+1}}\right) + \underbrace{1+\dots+1}_{k+1 \text{ terms}}$

How many times should we substitute to reach basic case $T(1)$?

k should be such that $\frac{n}{2^{k+1}} = 1 \iff k = \frac{\log(n)}{\log(2)} - 1 = \log_2(n) - 1$

So if we subs. $\log_2(n)-1$ times, $T(n) = \underbrace{1+\dots+1}_{\log_2(n) \text{ terms}} = \boxed{\log_2(n)}$

III - Max and Min

If you choose to use a divide and conquer approach on this problem you should get the same complexity that we had for the previous algorithm: $T(n) = \frac{3n}{2} - 2$. This is because only the basic case $n=2$ saves you comparisons as you compare only one time two numbers to get max and min. The other cases takes 2 comparisons as you compare 2 min and 2 max to get min and max:

$$\left.\begin{array}{l} \max T(n) = \max\left(\max T_{left}\left(\frac{n}{2}\right), \max T_{right}\left(\frac{n}{2}\right)\right) \\ \min T(n) = \min\left(\min T_{left}\left(\frac{n}{2}\right), \min T_{left}\left(\frac{n}{2}\right)\right) \end{array}\right\} \text{2 comparisons}$$