Tower of Hanoi

source dest

n tower_of_hanoi(n, from, to, aux)

{ if n == 1

1 cout<<move disk from source to dest

n-1 towerofHanoi(n-1, from, aux, to)

1 cout<<move disk from

n-1 towerofHan(n-1, aux, to, from)

}

Recurrence of T_o_H Recursive Algo

$$t(n) = 0 \quad , \text{ if } n=0$$

$$\cancel{= 1 \quad 2t(0)+1 \Rightarrow 2*0+1 \Rightarrow 1}$$

$$= 2t(n-1)+1 \quad \text{otherwise}$$

$$t(n) - 2t(n-1) = 1 \quad \text{..eq.1}$$

This is an example of a Inhomogenous recurrence.

(Non)

$$\underline{(x-2)}\ \underline{(x-1)}\ \ldots$$ Coming from RHS of eqn 1

Comes from left hand side of eqn 1

Roots of eqn pol^n are $\underline{1}$ and $\underline{2}$

both are multiplicity of 1 only

e.g.

r_1	2
r_2	3
r_3	3

General sol^n is of form

$$t(n) = c_1(1)^n + c_2(2)^n \ldots$$

$$c_1 + c_2 = 0, \quad n = 0$$

$$c_1 + 2c_2 = 1, \quad n = 1$$

From this, the sol^n $c_1 = -1$ & $c_2 = 1$

that's why

$$t(m) = c_1 1^m + c_2 2^m$$

$$= 2^m - 1$$

$$= 2^m - 1$$

ToH $t(m) \in \Theta(2^m)$

Book source:
By Brassard & Bratley

Change of variable

Complicated recurrences may be solved by making change of variable.

Example from book cormen pg - 86

$$T(n) = 2T(\lfloor \sqrt{n} \rfloor) + \lg n \quad \text{eq 1}$$

$\therefore \sqrt{n} \Rightarrow n^{1/2}$

$\therefore (2^m)^{1/2} \Rightarrow 2^{m/2}$

Let rename $m = \lg n$ for convenience

$\therefore 2^m = 2^{\log_2 n}$

$= n^{\log_2 2}$

$= n \cdot 1 = n$

$$T(2^m) = 2T(2^{m/2}) + m \quad \text{eq. 2}$$

We can now rename

$$S(m) = T(2^m) \quad \text{in eq } 2 \therefore$$

$$S(m) = 2S(m/2) + m$$ ——— eq. 3

$T(2^{m/2})$

$T($

$T(n) = aT(n/b) + f(n)$ ——— ∴ apply Important formula.

so $O(m \lg m)$ ∴ case 2 of master method

$\therefore\ n^{\log_b a}$ ‖ $f(n)$ m

$m^{\log_2 2}$ ‖ m

$n = 2^m$

$m = \lg n$ $\rightarrow O(\lg n \cdot \lg \lg n)$

Finding maximum and minimum from an array

max = min = a[0] | duplicates?

for i = 1 to n-1

{

if (a[i] > max)

max = a[i]

else if (a[i] < min)

min = a[i]

}

max

min

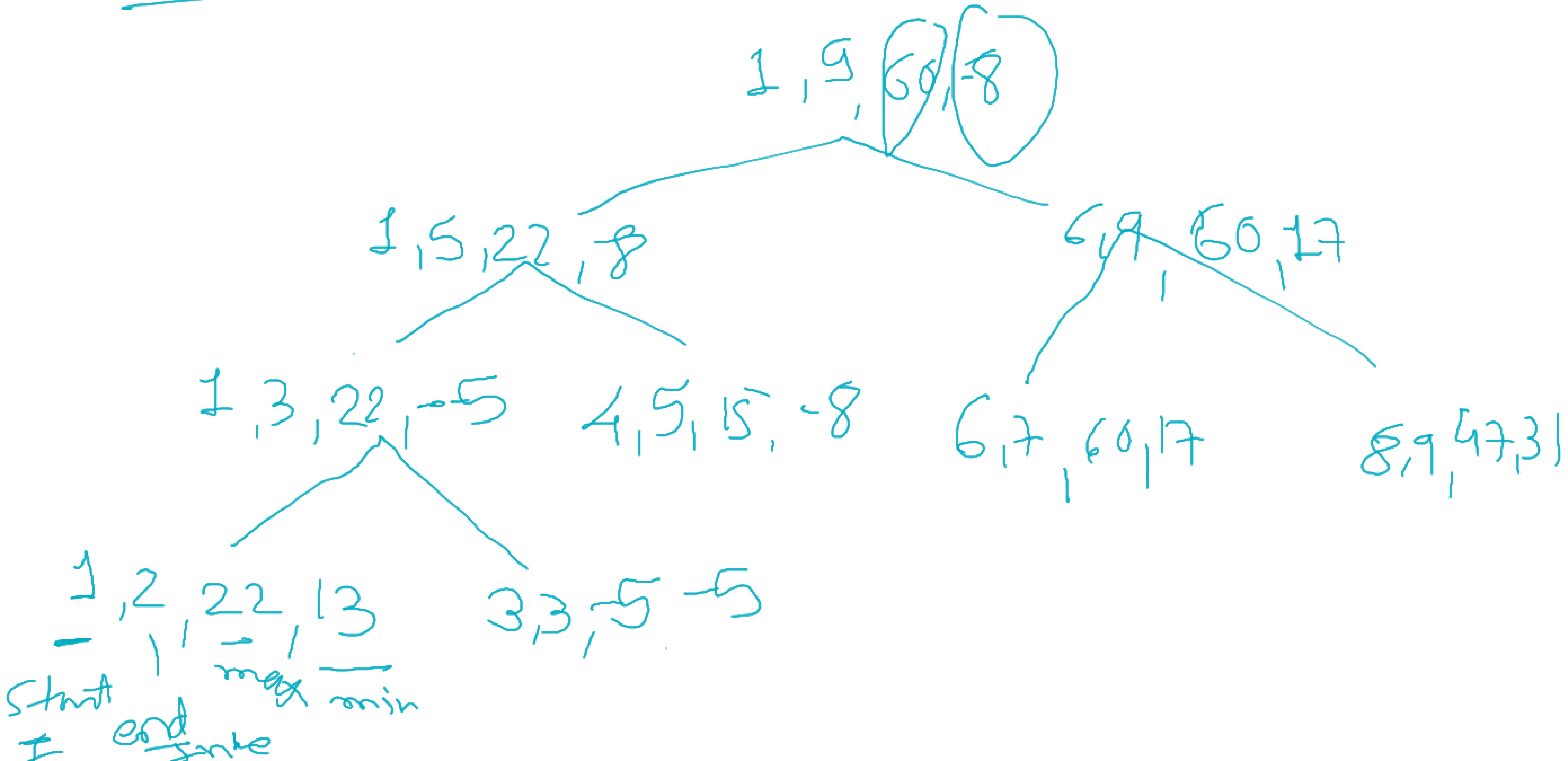
Iterative

Author says

2n-2

comparisons Ans

22 13 -5 -8 15 60 17 31 47 $n = 9$



$$T(n) = 2T(n/2) + 2$$
$$= 2(2T(n/2) + 2) + 2$$
$$= 4T(n/4) + 4 + 2$$
$$= \vdots$$
$$= 2^{k-1}T(2) + \sum_{1 \le i \le k-1} 2^{i}$$
$$= 2^{k-1} + 2^{k} - 2 = 3n/2 - 2$$

Algorithm analysis of recursive max min

$$3\frac{n}{2} - 2$$

Vs iterative

$$2n - 2$$

25% of benefit.

But we are gonna need lot of space requirement which is overhead of recursive max min