Divide and Conquer

Finding Max and Min using D&C approach

Problem — To find max and min value in an array.

Naive method —

Algo : max-min(a[ ])

```
max = a[1]
min = a[1]
for i = 2 to n do
    if a[i] > max then
        max = a[i]
    if a[i] < min then
        min = a[i]
return (max, min)
```

No. of comparison = $2n - 2$

⇒ No. of comparison can be reduced using divide and conquer approach

```
Algo  RMaxMin(i, j, max, min)
{
if (i==j) then  max = min = a[j]        -(1)
else if (i==j-1) then
    {
        if(a[i] < a[j]) then
            {
            max=a[j]; min=a[i]; }
        else
            { max=a[i]; min=a[j];}
    }                                    (2)
else
    {
    mid = ⌊(i+j)/2⌋                      -(3)
T(n/2) - RMaxMin(i, mid, max, min);      } conquer
T(n/2) - RMaxMin(mid+1, j, max1, min1);  } conquer
    if(max < max1) then  max = max1;     } combine
    if(min > min1) then  min = min1;
}
}
```

Divide.

22 Sunday

Example –

Index

1	2	3	4	5	6	7	8	9
22	13	-5	-8	15	60	17	31	47

i ↑ (index 1) j ↑ (index 9)

i is location of first element = 1
j is location of last element = 9

① if(i==j) // means single element in array then max=min will be same. This is a small problem.

② if(i==j-1) means suppose array has 2 elements

1	2
22	13
i	j

i==j-1

Then in that case compare both and assign max and min accordingly.
This is also a small problem.

Else if more than 2 elements then ③ case applies.

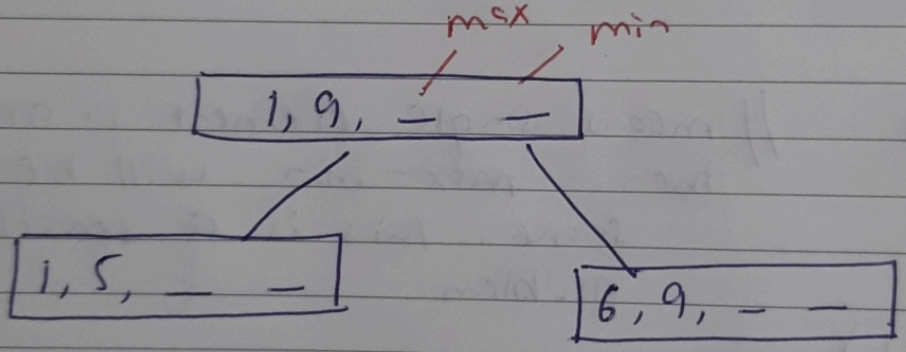
(3) When more than 2 elements in an array

It is a big problem to be solved using recursion.

mid = ~~(i+j)/2~~ $\lfloor (i+j)/2 \rfloor$

mid = $\lfloor (1+9)/2 \rfloor = \lfloor 5 \rfloor = 5$

↑
index



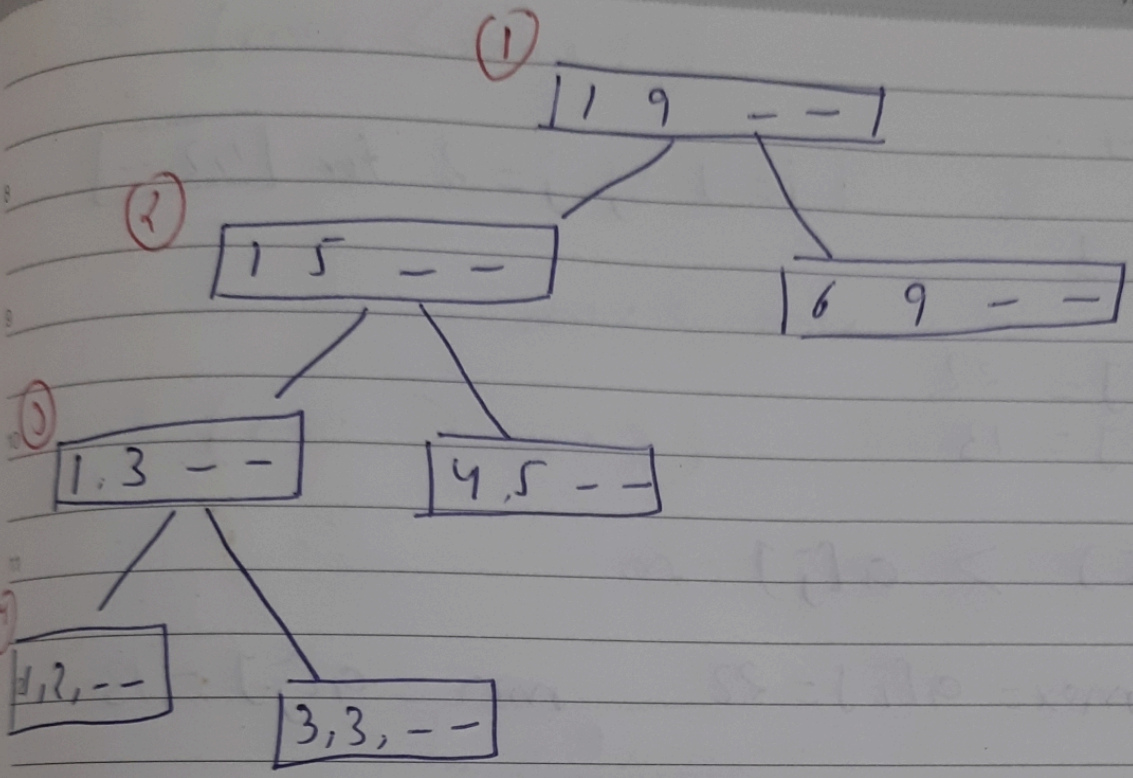
1st call.

~~Rmaxmin(i, mid, max, min)~~

Rmaxmin(mid+1, j, max, min)

First left subtree will be divided

Then right subtree will be divided.



(1) $i = 1$, $j = 9$ $mid = 5$

(2) $i = 1$, $mid = 5$ $mid+1 = 6$ $j = 9$

(3) case 3

$i = 1$ $mid = (5+1)/2 = 3$ $mid+1 = 4$, $j = 5$

1,3 - -

4,5, - -

(4) case 3

$i = 1$ $mid = (3+1)/2 = 2.$, $j = 3$ $mid+1 = 3$

1,2 - -

3, 3, - -

After step 4 — $i = 1$, $j = 2$ for 1,2,--

$i == j - 1$

$\therefore$ $a[1] = 22$
$a[2] = 13$

if $a[i] > a[j]$

$\therefore$ $max = a[i] = 22$ $\quad min = a[j] = 13$

$\therefore$ 1, 2, 22, 13

max min.

Similarly for second node 3,3,--

$i == j$ then $max = min = a[j]$

$a[3] = -5$

$\therefore$ ~~1, 2~~ 3, 3, -5, -5

max1 min1.

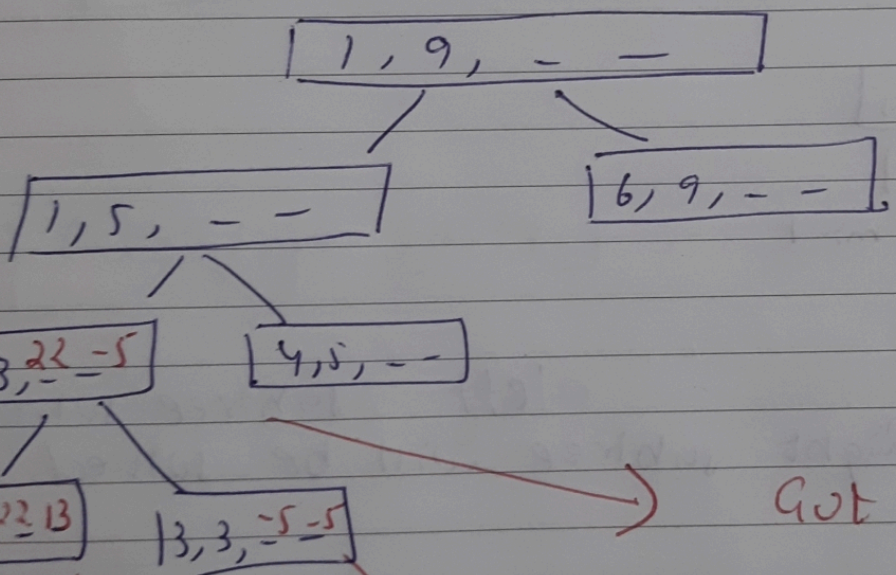
if (max < max1)

22 < -5 X

∴ max = max = 22

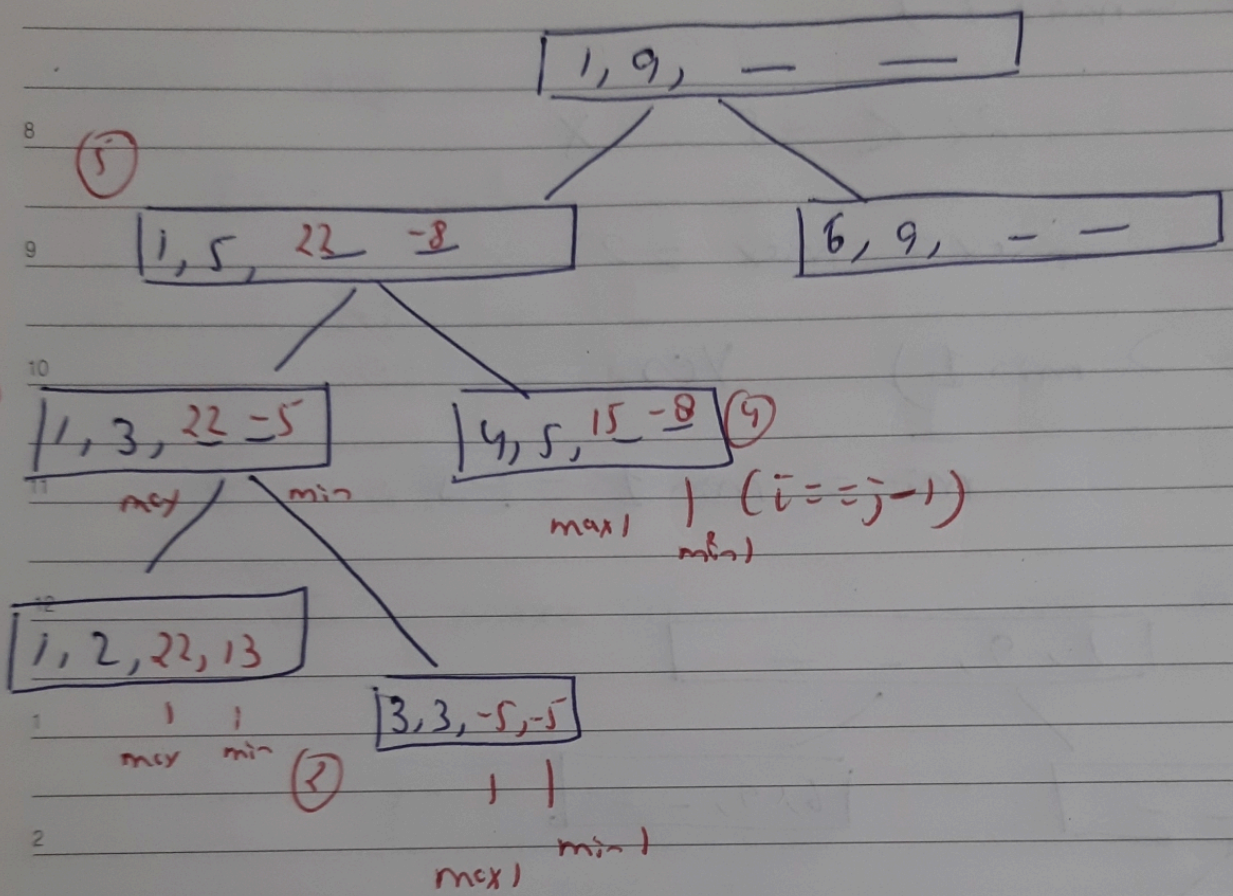
if (min > min1) Yes

∴ min = min1 = -5



Got max, min for (3) step

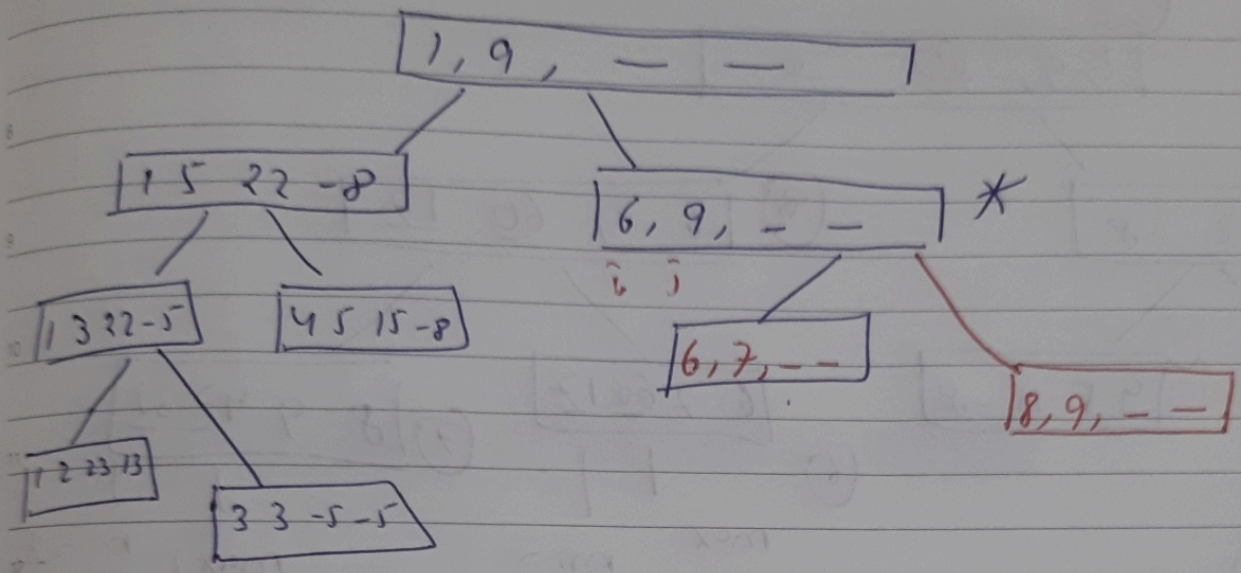
Similarly trace for upwards.



left subtree solved.
Now Right subtree will be solved.

29 Sunday

2015



* ① $\lfloor (6+9)/2 \rfloor = \lfloor 15/2 \rfloor = \lfloor 7.5 \rfloor = 7$. floor

$i = 6 \quad j = 9$ mid = 7, mid + 1 = 8

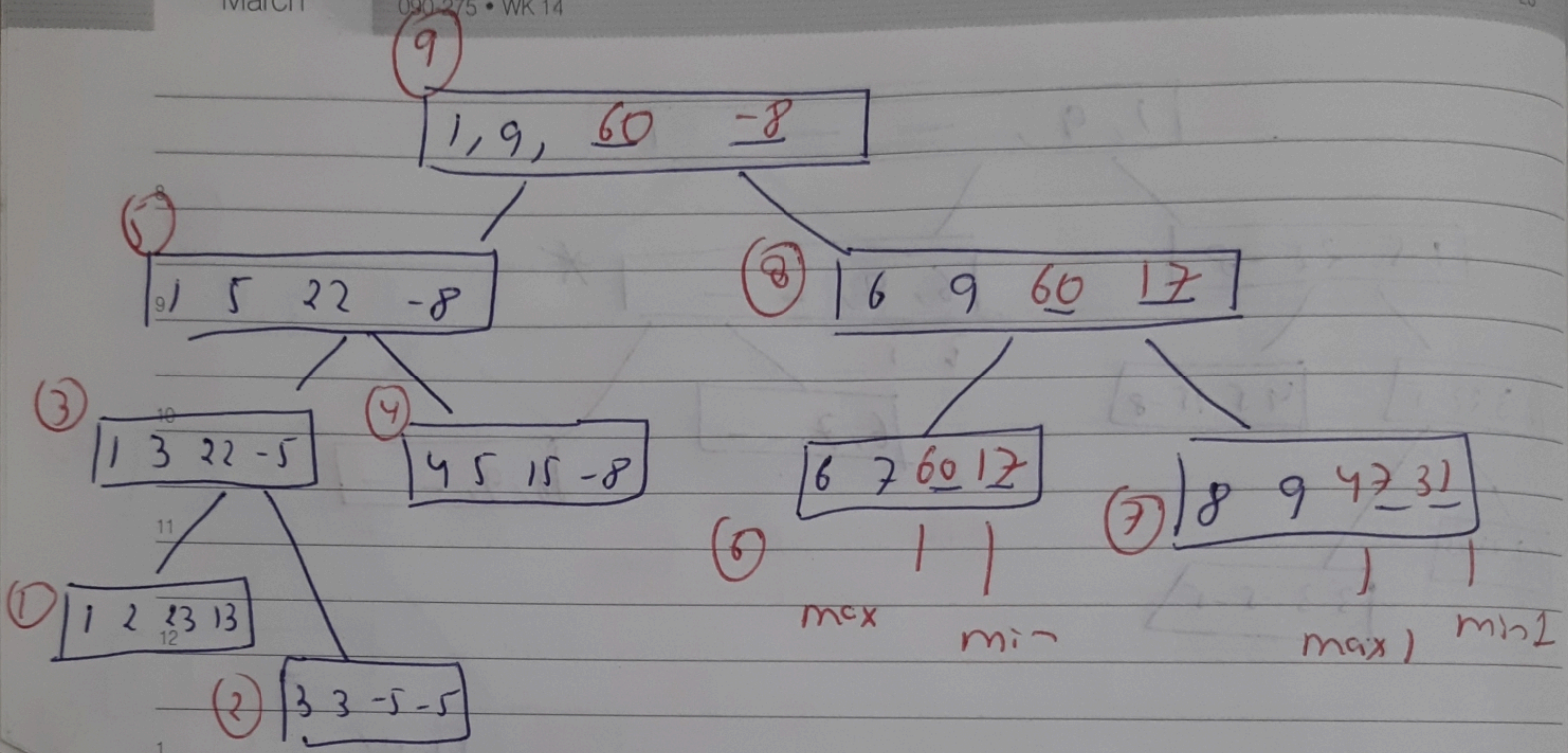
② 6, 7, — — $i == j - 1$ case 2 applies

$\therefore$ max = 60, min = 17

Similarly for 8, 9, — — case 2 applies

$i == j - 1$

$\therefore$ max1 = 47, min1 = 31



In (9) node => (22, -8) (60, 17)
max min max1 min1

will be compared to get

max of all elements = 60

min of all elements = -8 Ans.

Wednesday
April

091-274 • WK 14

Time complexity

(1) case 1 takes constant time (1) time

(2) case 2 also takes constant time (1) time

(3) ~~case 3~~ All small problems take constant time.

(3) case 3 is recursive call.

mid calculation takes constant time

1st recursive call will take $T(n/2)$ as for half elements call is made

2nd recursive call also takes $T(n/2)$

condition checking takes constant time

$$\therefore T(n) = \begin{cases} 1 & n = 1, 2 \\ 2T(n/2) + 1 & n > 2 \end{cases}$$

Solving using masters theorem – Case 1.

$T(n) = O(n)$