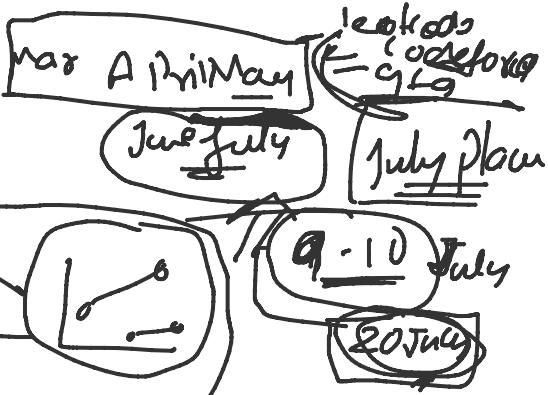
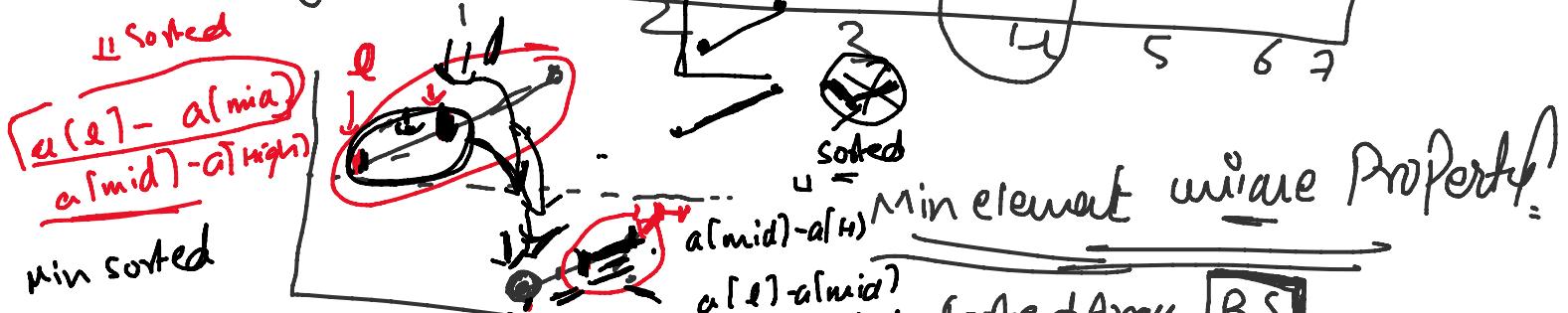
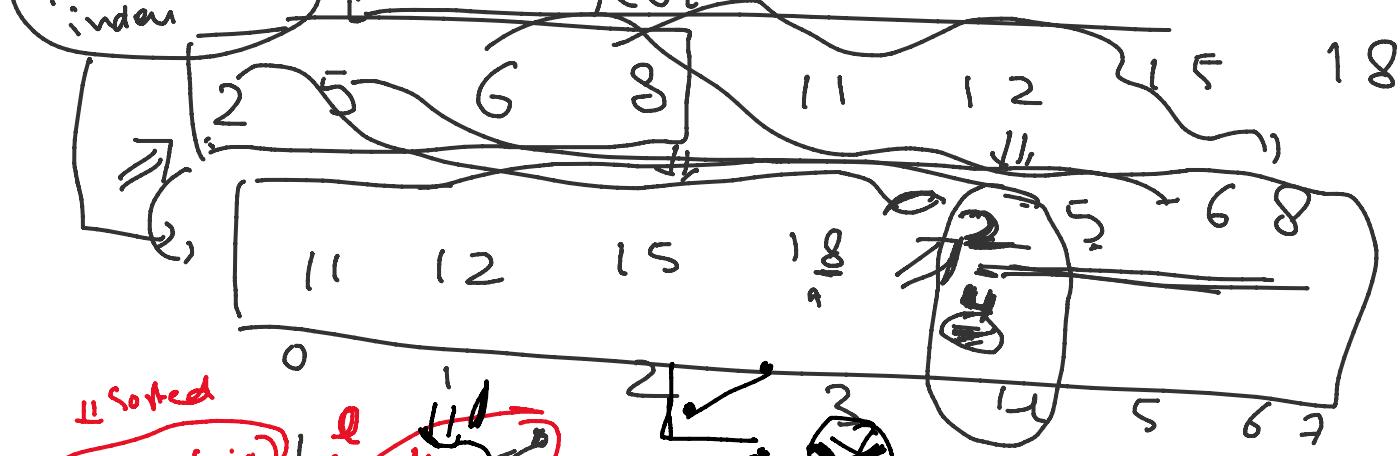


CLASS - 16Binary Search

No of Times a sorted array is Rotated //



1) Mid element == requirement

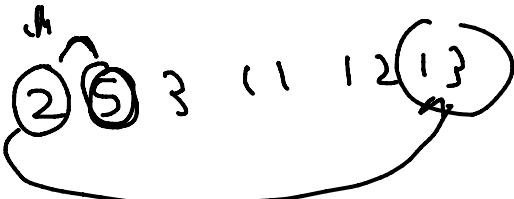
Sorted Array BS  
① Search Space

2) if (mid != require)  $a[prev] > a[mid] < a[next]$

⇒ Search Space reduce

to reach our answer closer

if ( $a[0] < a[n-1]$ )  
int l = 0;  
int h = n - 1;

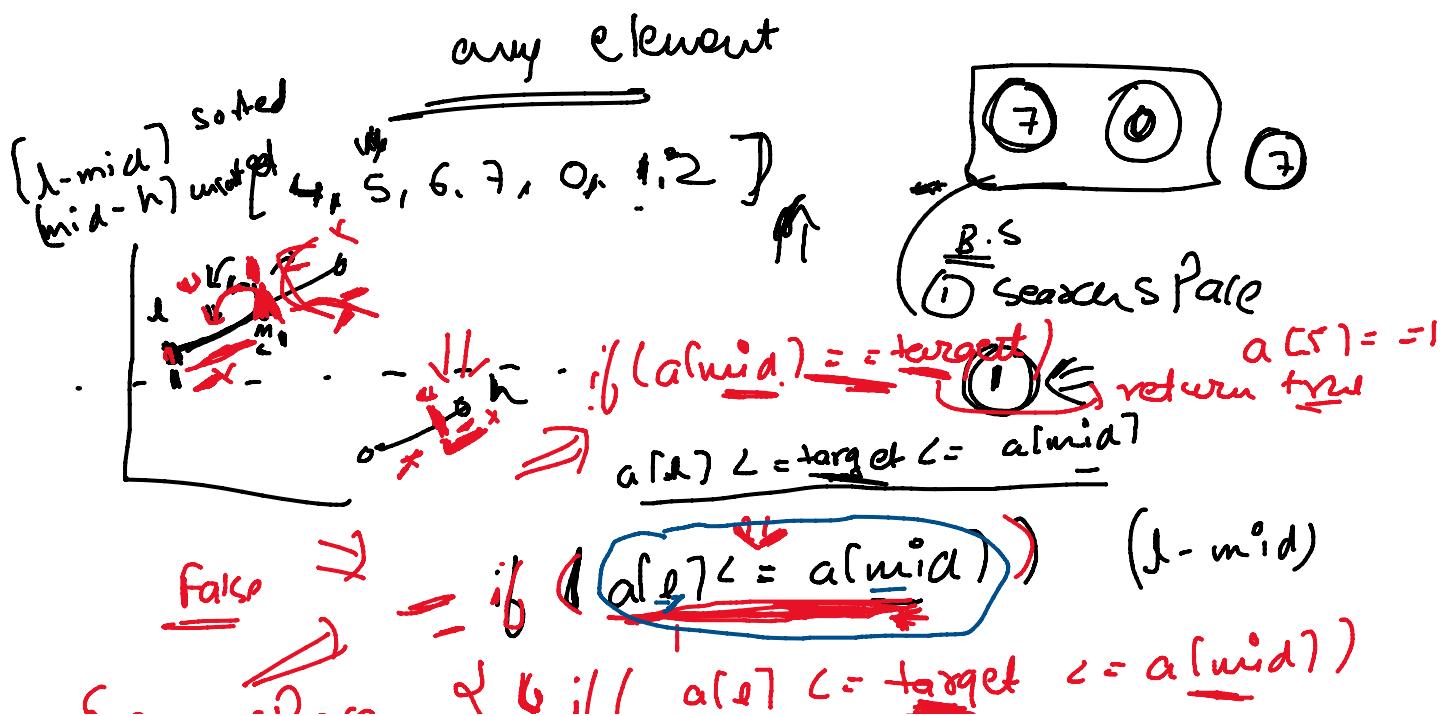


```

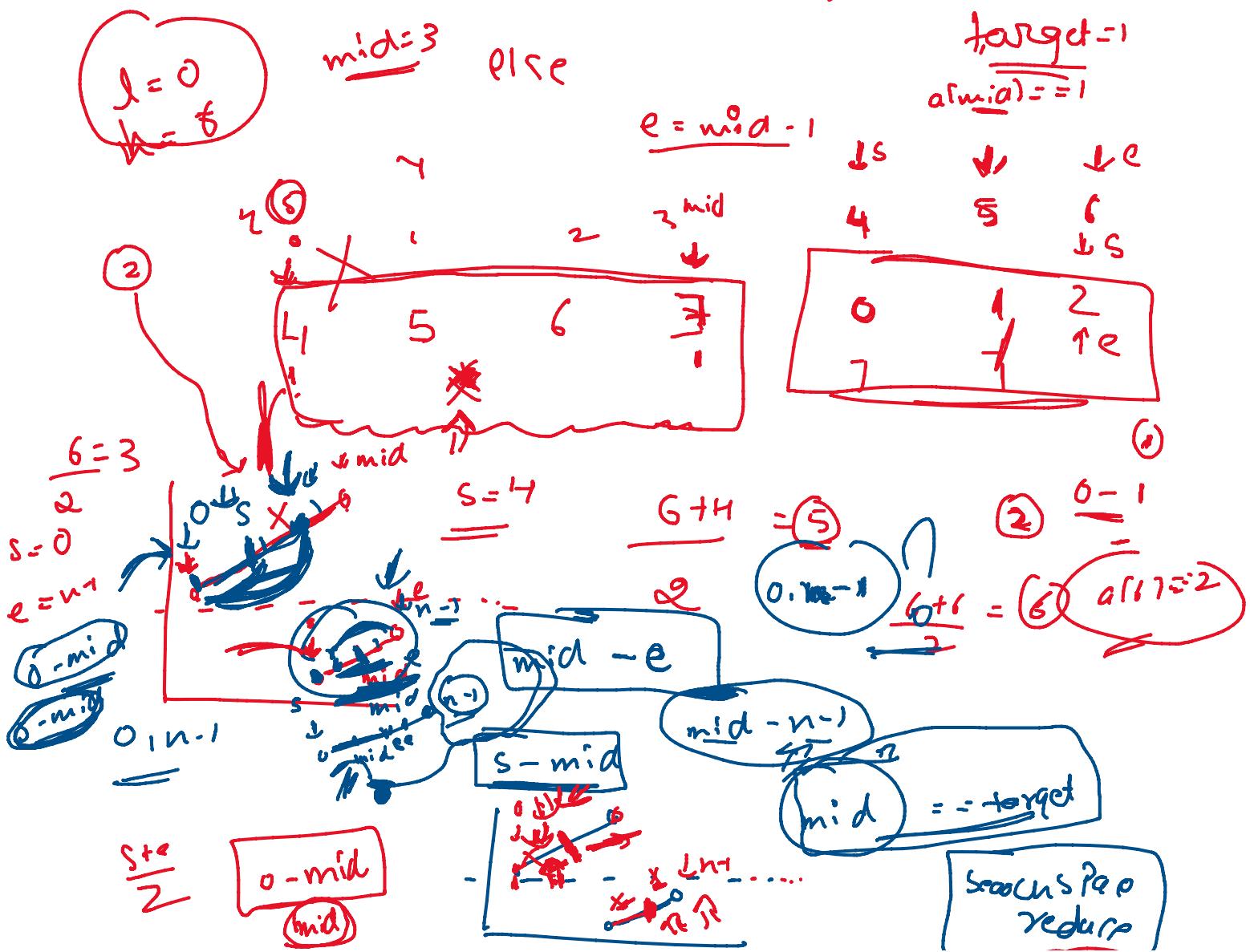
    'vl un'
    $ return 0;
    ~~~~~~
    int H = n - 1; // n - 3
    while (l <= H) {
        int mid = l + (H - l) / 2; //  $\left\lfloor \frac{H+l}{2} \right\rfloor$ 
        int Prev = (mid - 1 + n) % n;
        int next = (mid + 1) % n;
        mid == next if (a[mid] <= a[prev] and a[mid] <= a[next])
        "we got the min element."
        getagen null;
    }
    else if (a[mid] <= a[l]) [mid >= n]
        h = mid - 1;
    else if (a[mid] >= a[l])
        l = mid + 1;
    }

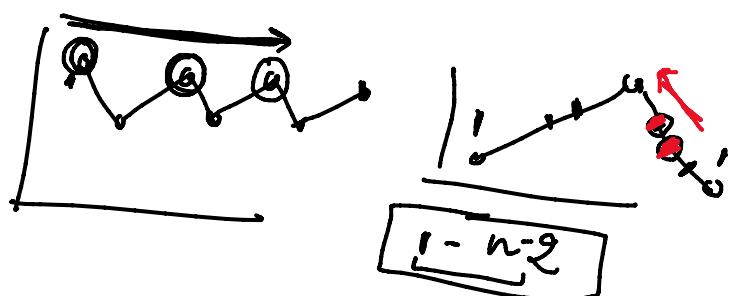
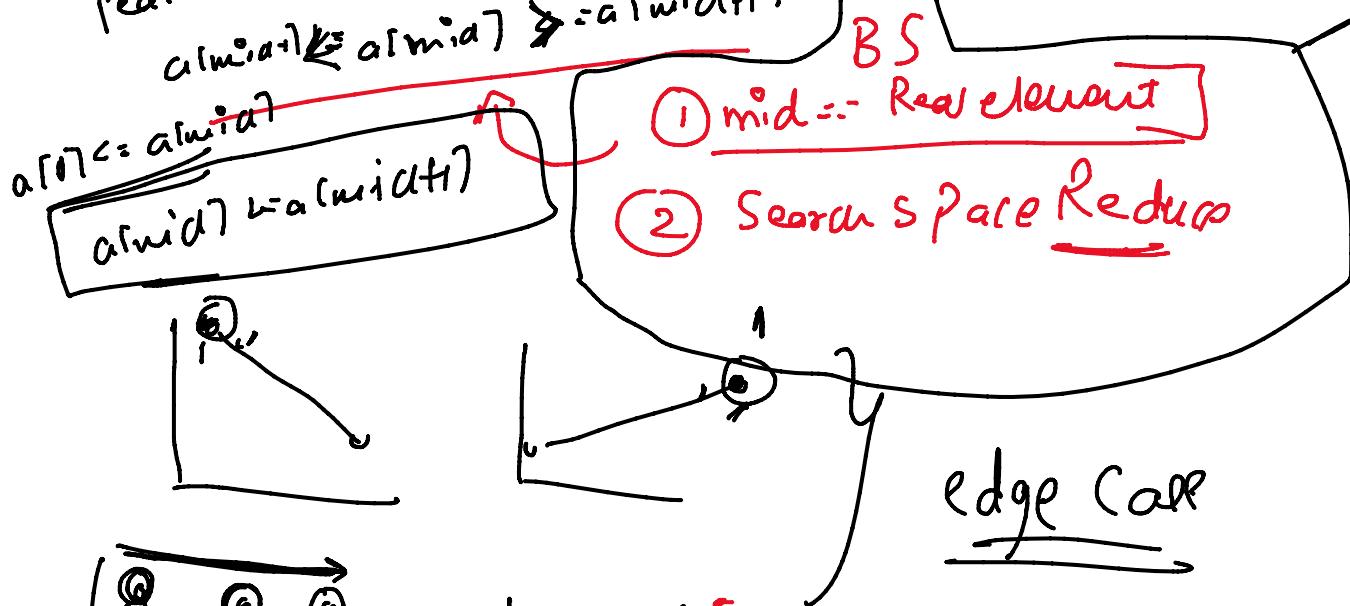
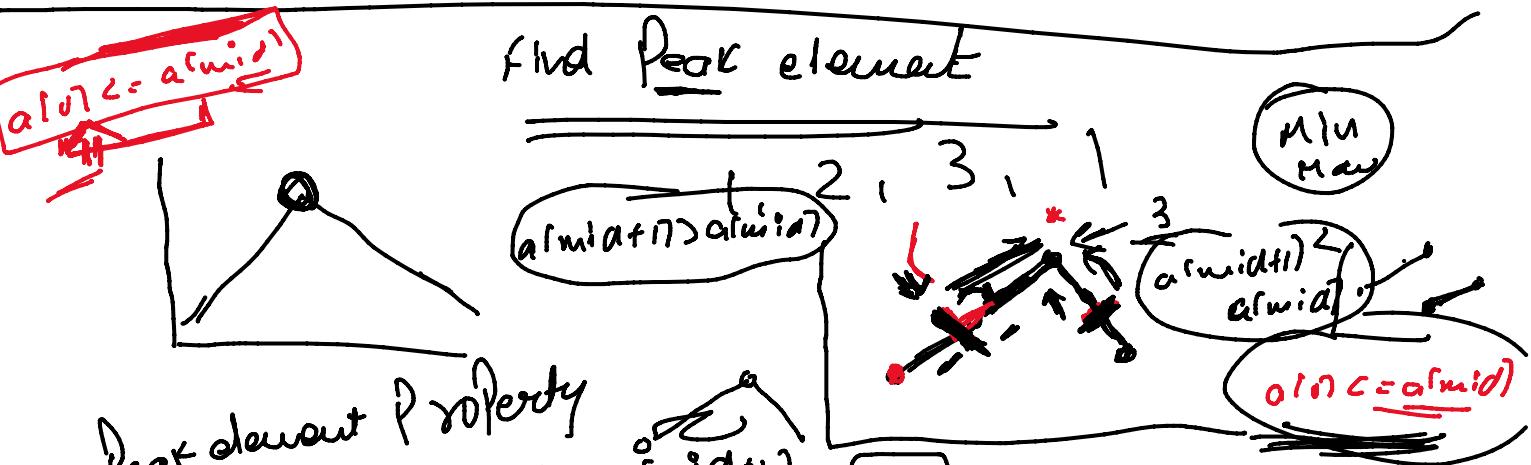
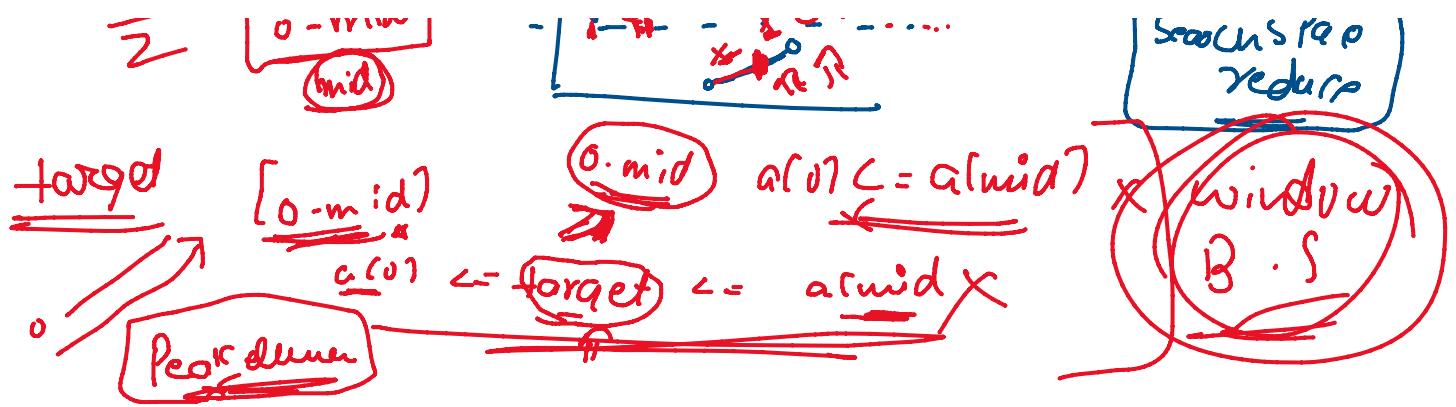
```

Sorted rotated array  $\rightarrow$  Min element



SearchSpan  
 if ( $a[s] < target$   $\leftarrow a[mid]$ )  
 $e = mid - 1$   
 else  
 $s = mid + 1$   
 else if ( $a[mid] < target$   $\leftarrow a[h]$ )  
 $s = mid + 1$





Peak\_element X

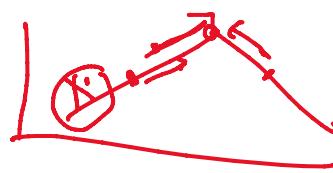
$\{ \text{end } n-1 \}$

1 2 3 4 5 2 1

1 2 3 4 5 ↙ |

Decide

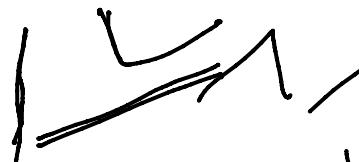
Search Space



Binary Search

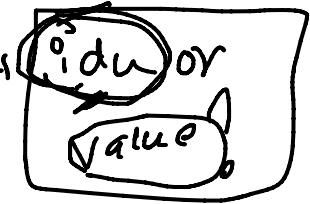
→ Min  
→ Max  
→ target

value



idx

Binary search  
 $a[idx] = \underline{\text{value}}$



Binary Search on A[ ]



min page

Max Page



Next student  
Next Page = 45

B.S.  $\Rightarrow$  2nd Category

Minimize or Maximize

$10 + 20 + 15 + 30 = 75$

75

Min Page  $\Rightarrow$  0

65 Max Page  $\Rightarrow$  75

$10 + 20$

30

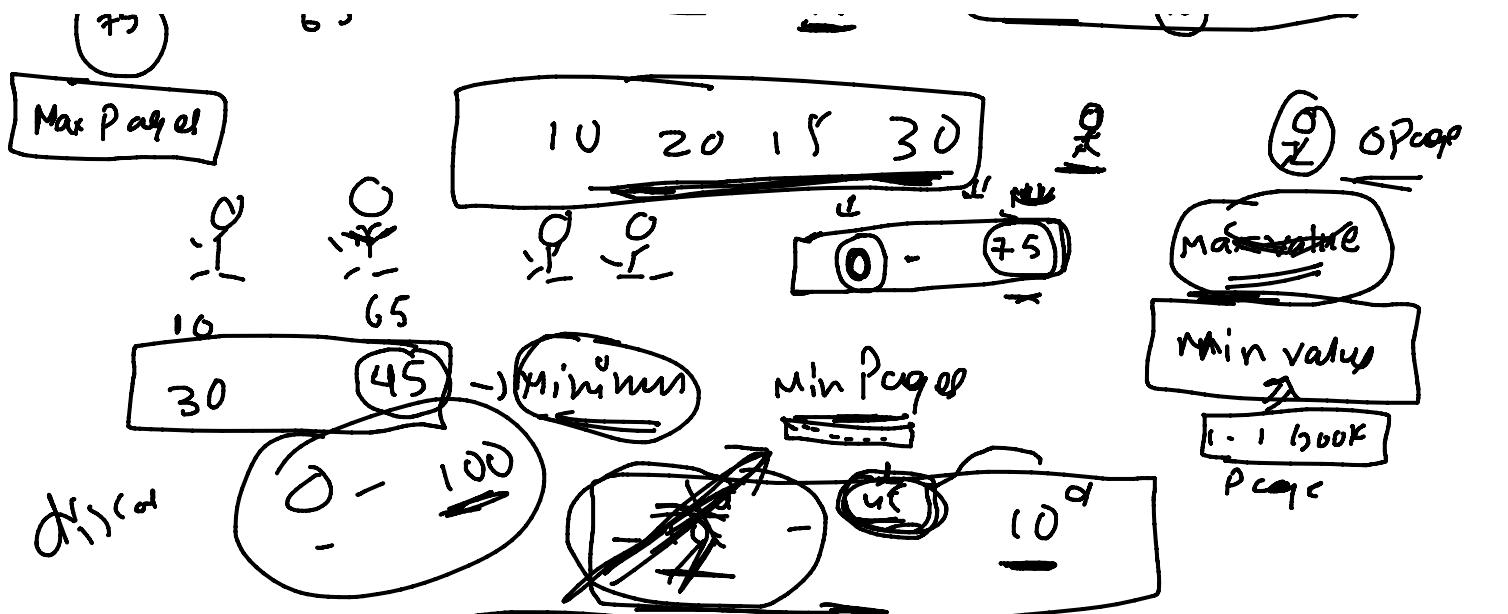
10 + 20

30 + 15

45

0 - 75 Page  
70

76x



$$0 = - = - 75$$

45

