

$l = 0$   $l > r$   $r = n - 1$

$l == r$



(1)

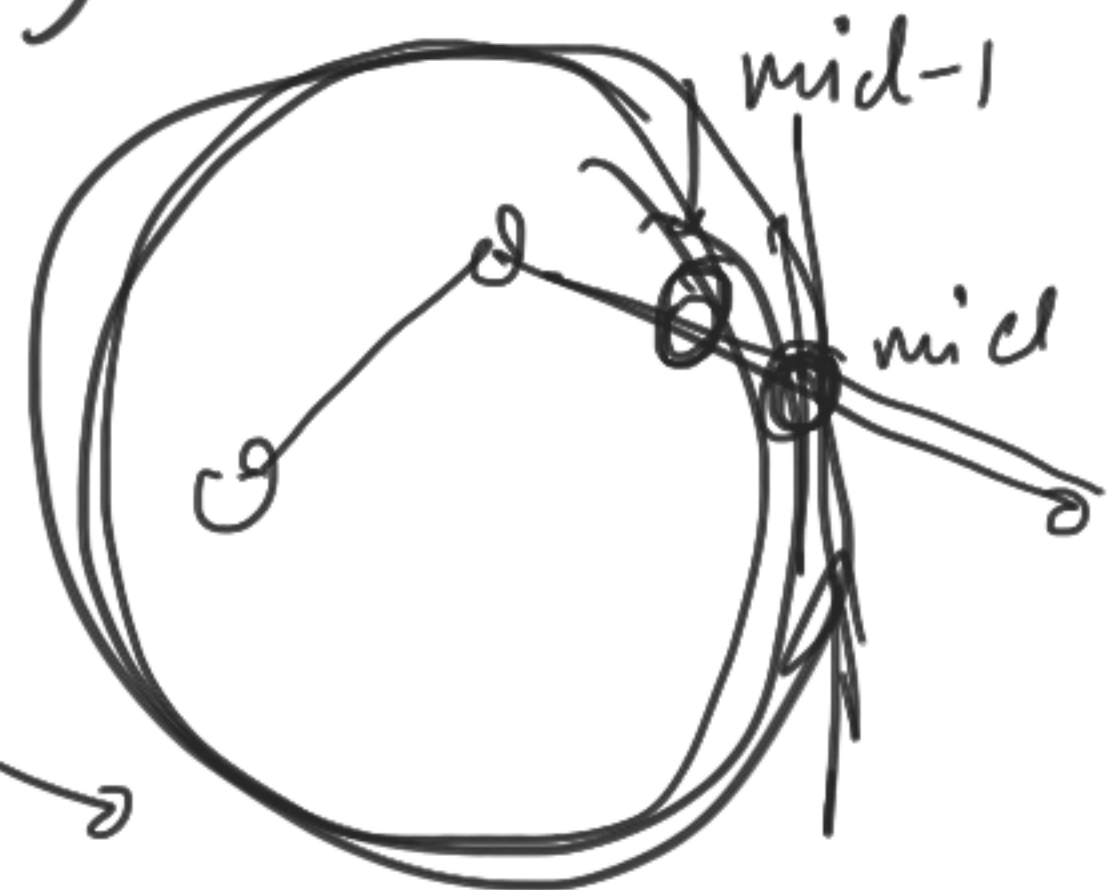
$\text{while } (l < r)$

$\text{if } (A[\text{mid}] < A[\text{mid} - 1])$

$r = \text{mid}$

$\text{else } l = \text{mid} + 1$

$\text{return } A[l] \text{ or } A[r]$



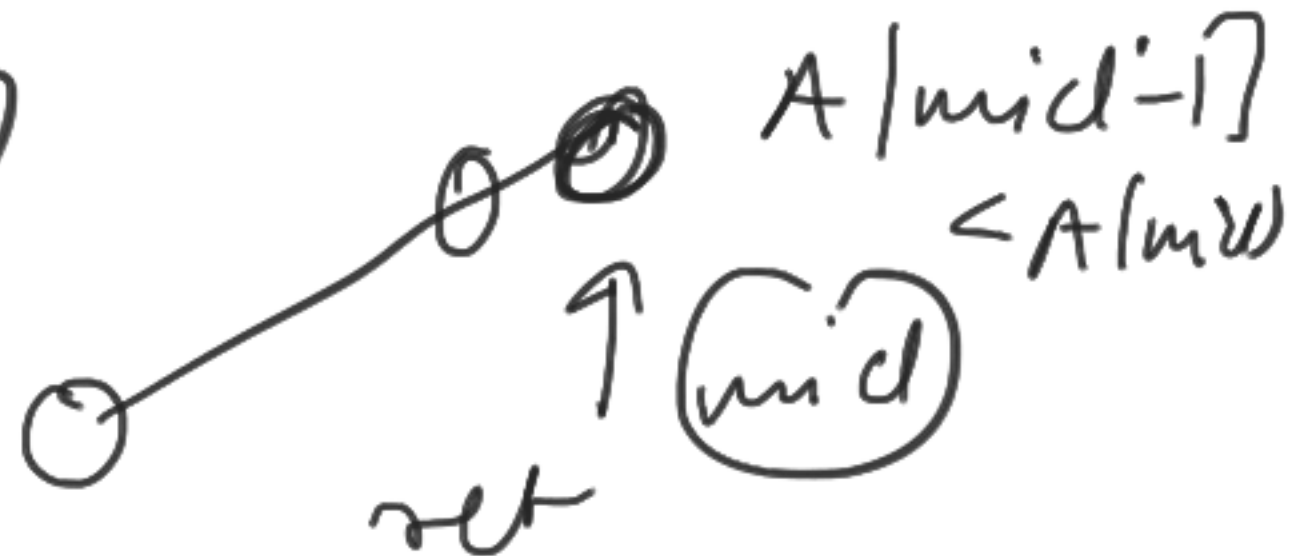
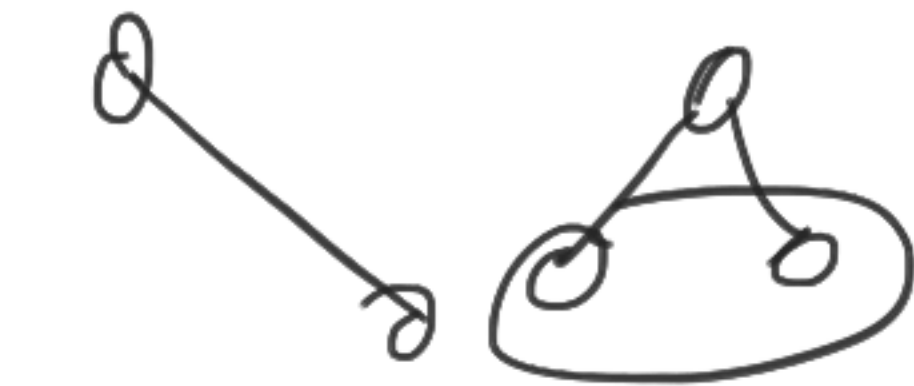
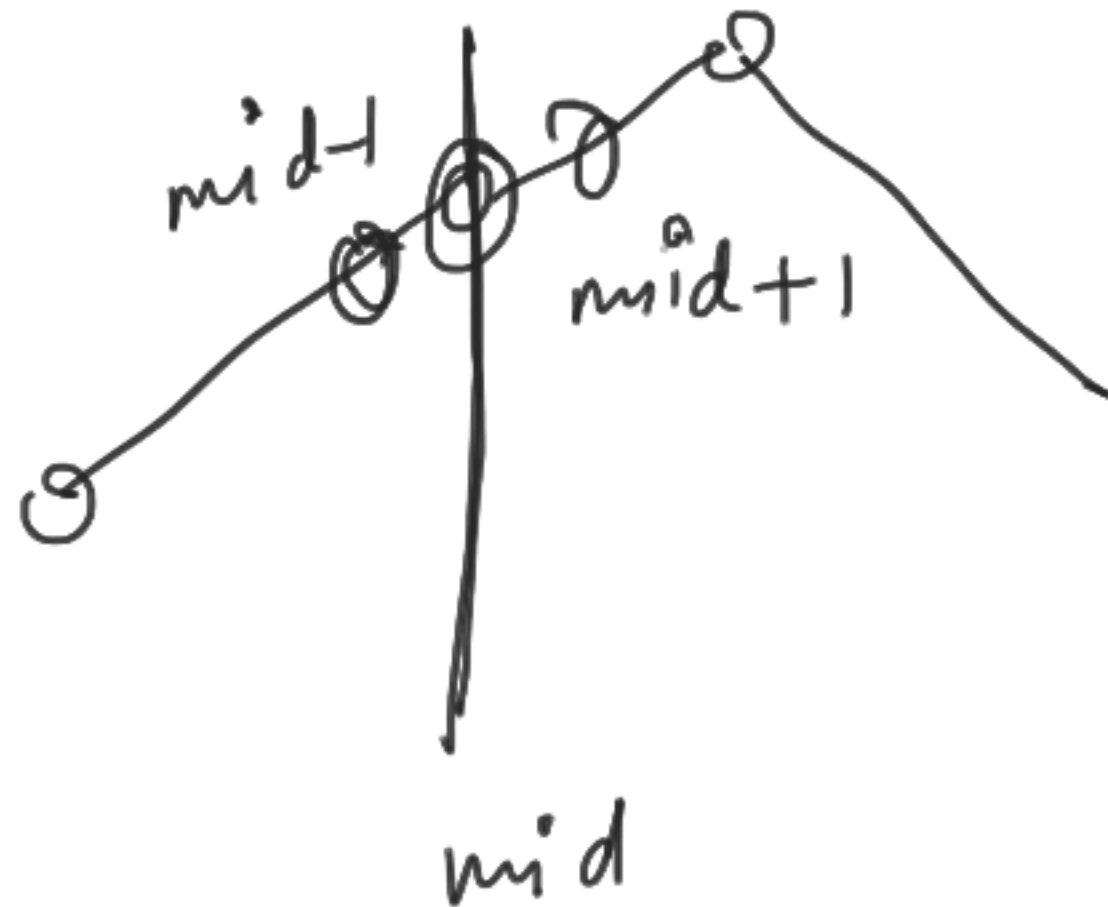
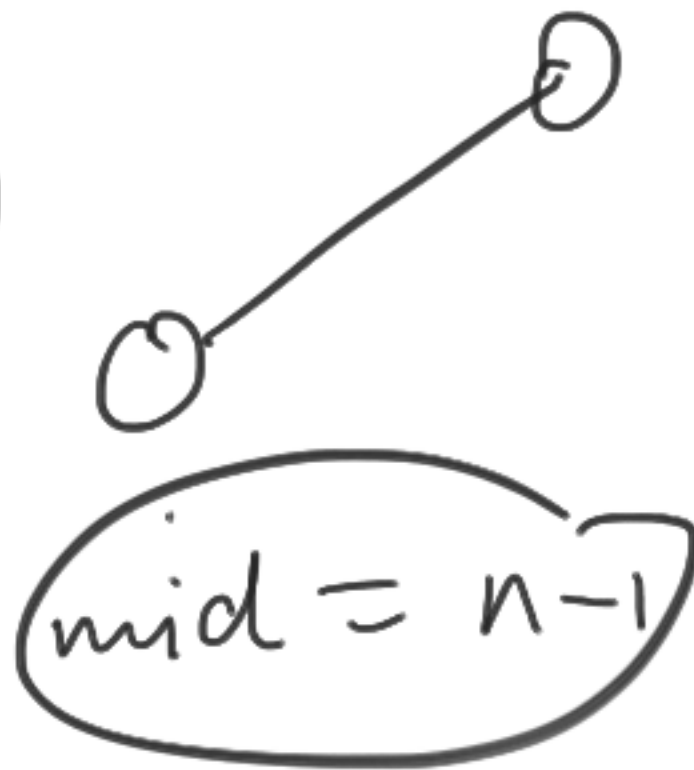
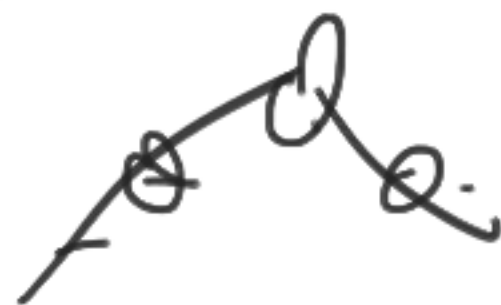
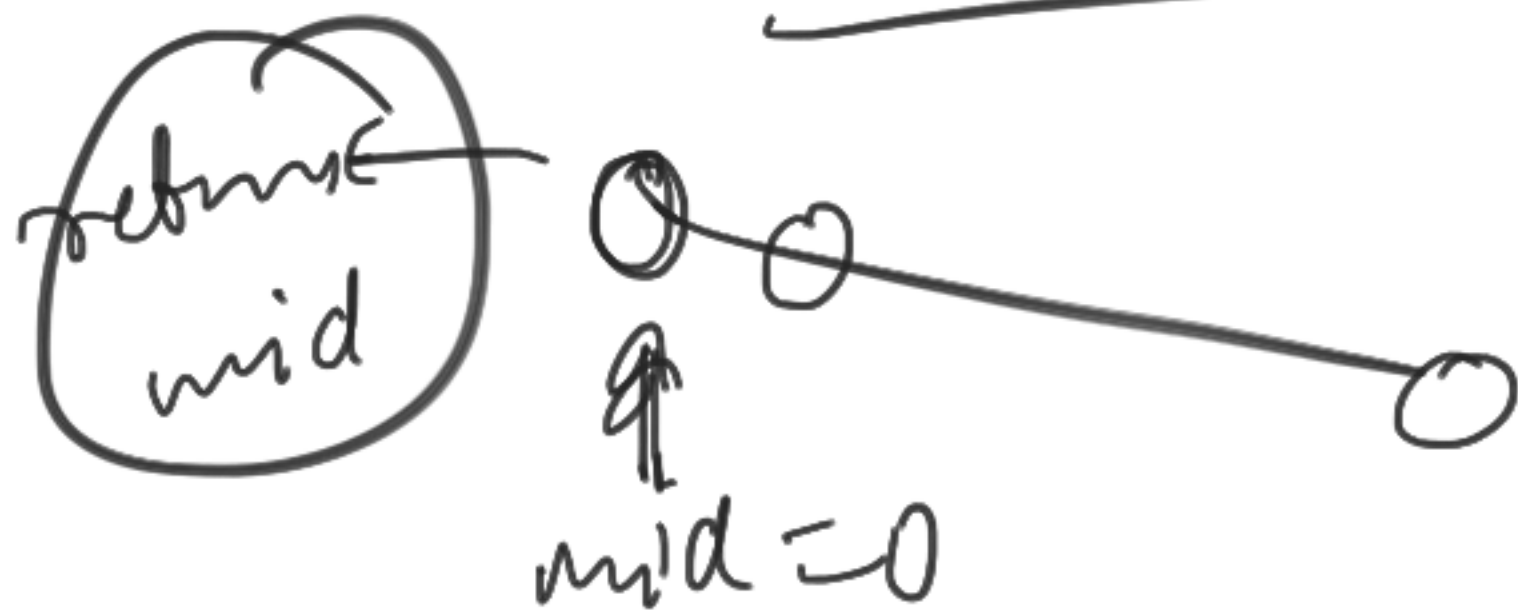
while ( )

{

return mid

mid = 0

$A[mid+1] < A[mid]$



if ( mid == 0 ||  $A[mid-1] < A[mid]$  )  
    ||  $(mid == n-1)$  ||

return mid.

else if ( mid > 0 &&  $A[mid-1] > A[mid]$  )

→ go to left side  $r = mid - 1$

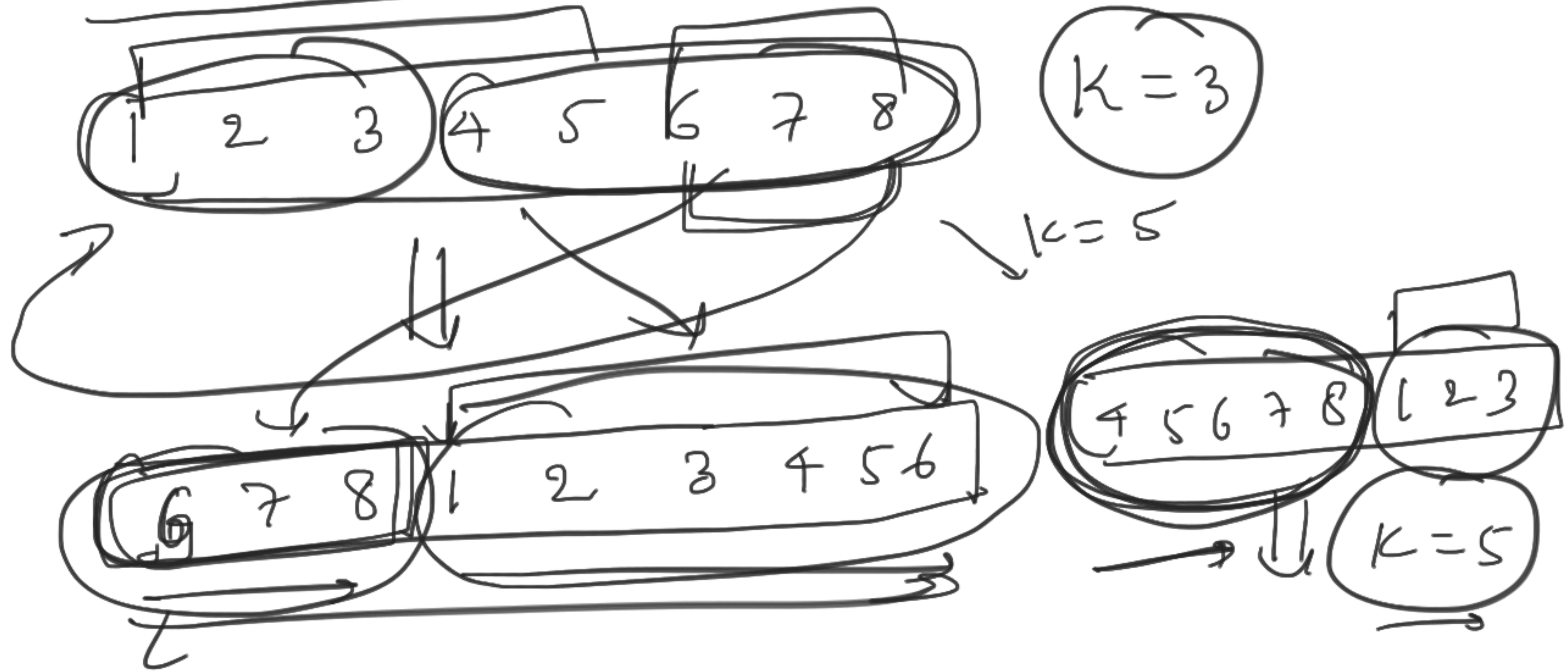
else

→ go to the right side

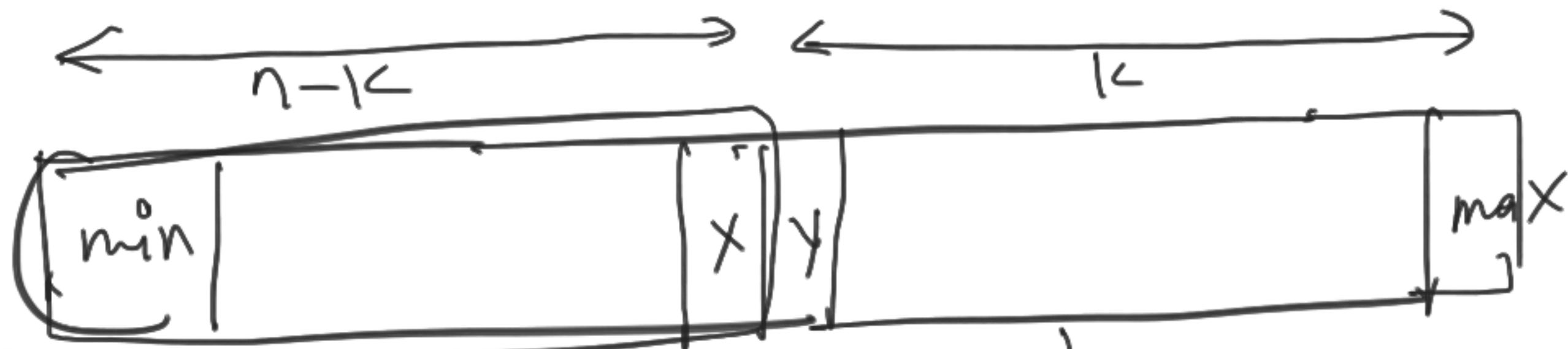
}

$d = mid + 1$

find min sorted and Rotated array.



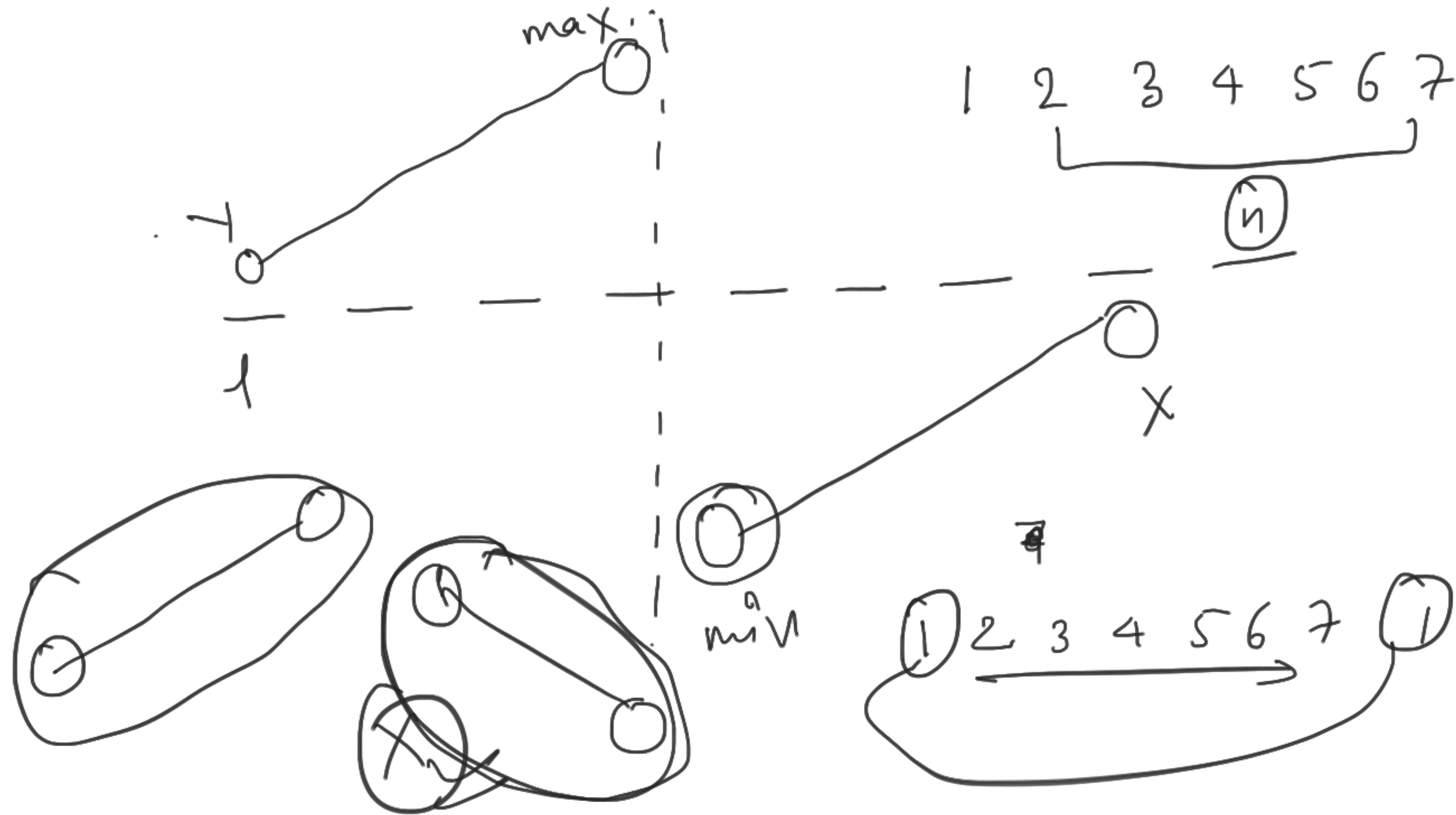
$a_{min}$   
 $< x < y < max$



$max > y > x > min$

$n-k$

Sorted



$A[mid]$

$l$

$r$

$mid$

$min$

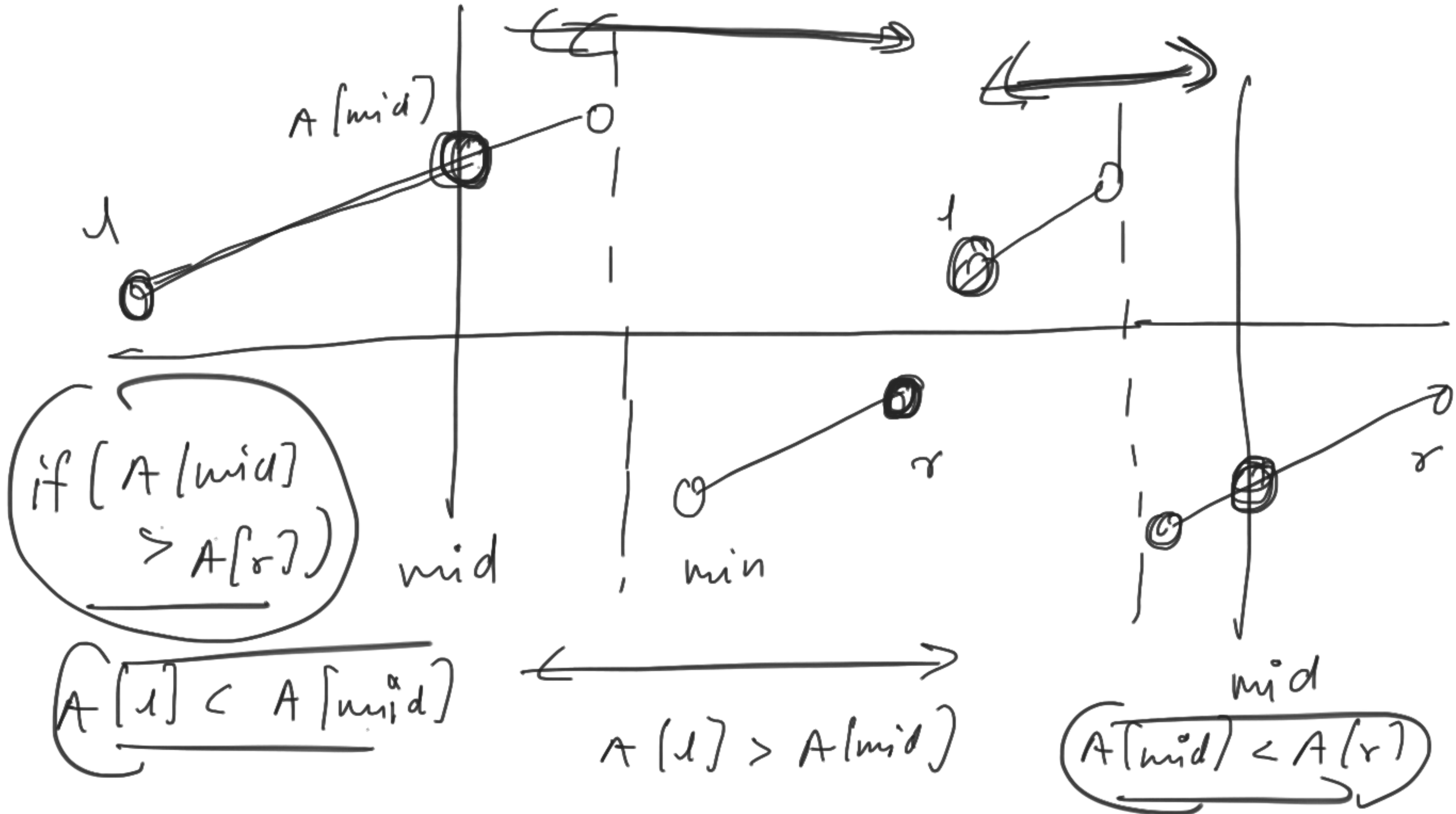
$mid$

if  $(A[mid] > A[r])$

$A[l] < A[mid]$

$A[l] > A[mid]$

$A[mid] < A[r]$



$l = 0, r = n - 1$   $(l = r)$

while  $(l < r)$

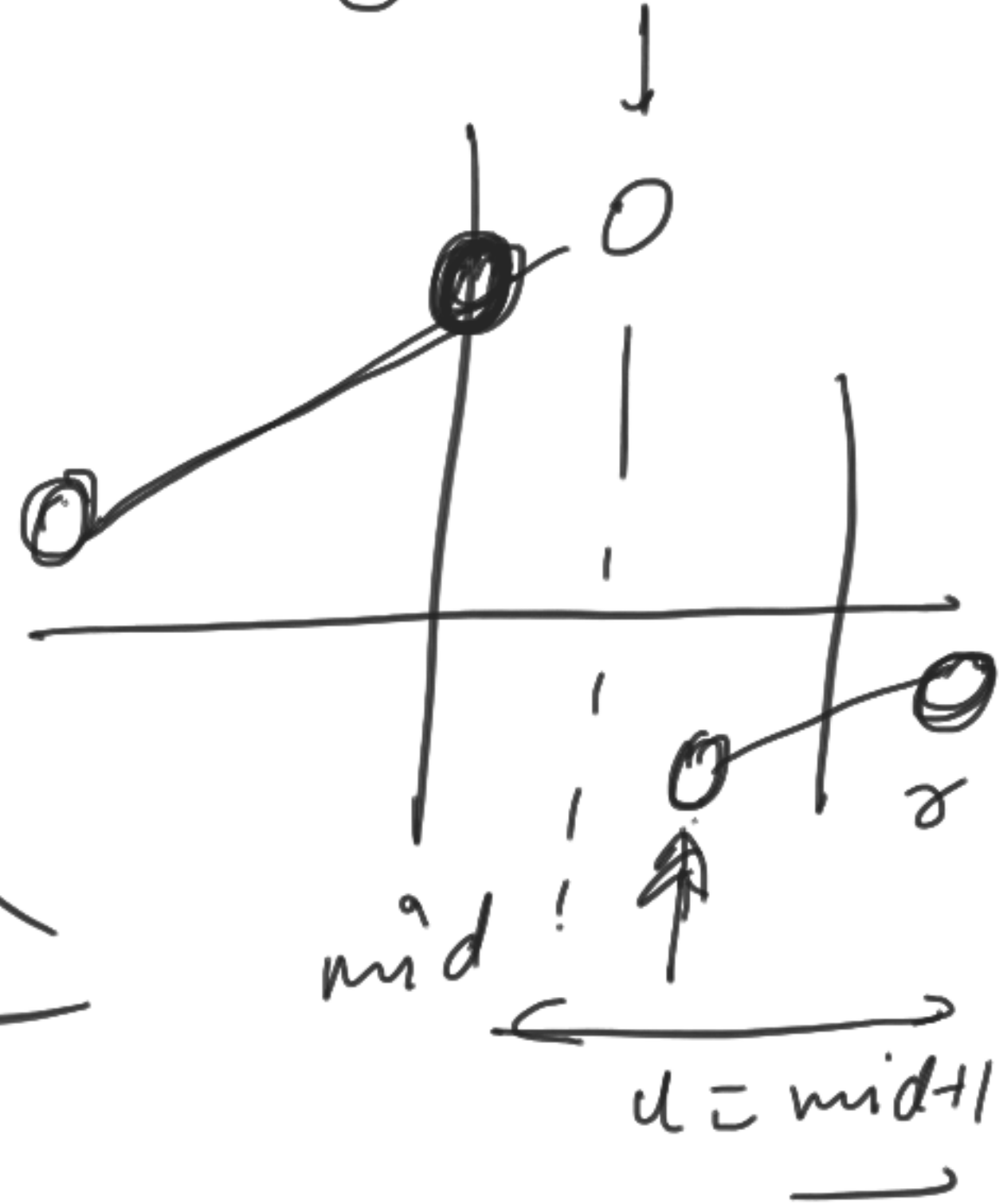
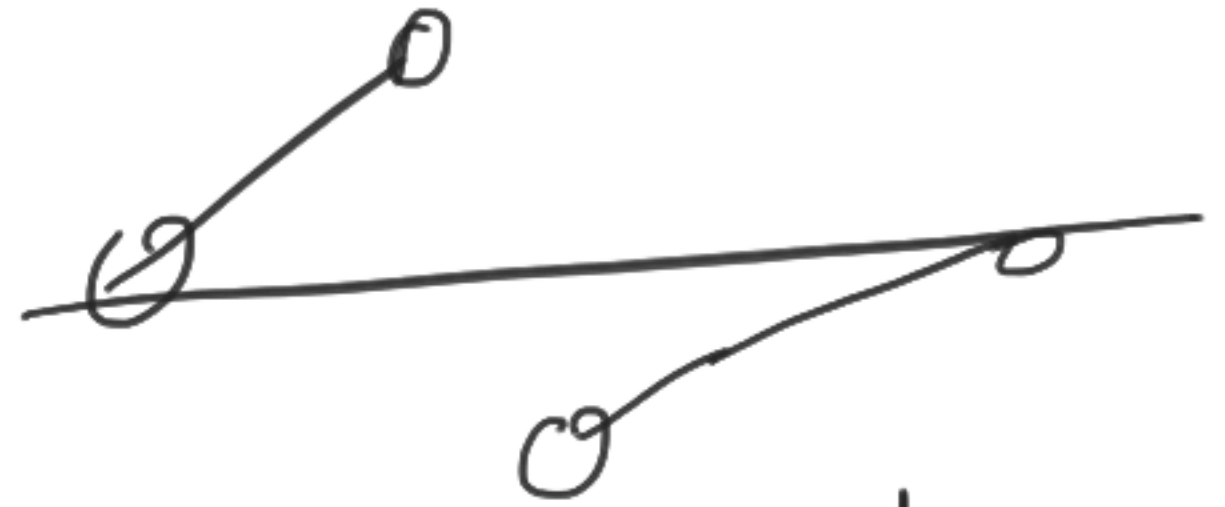
{ if  $(A[l] > A[r])$

$l = mid + 1$

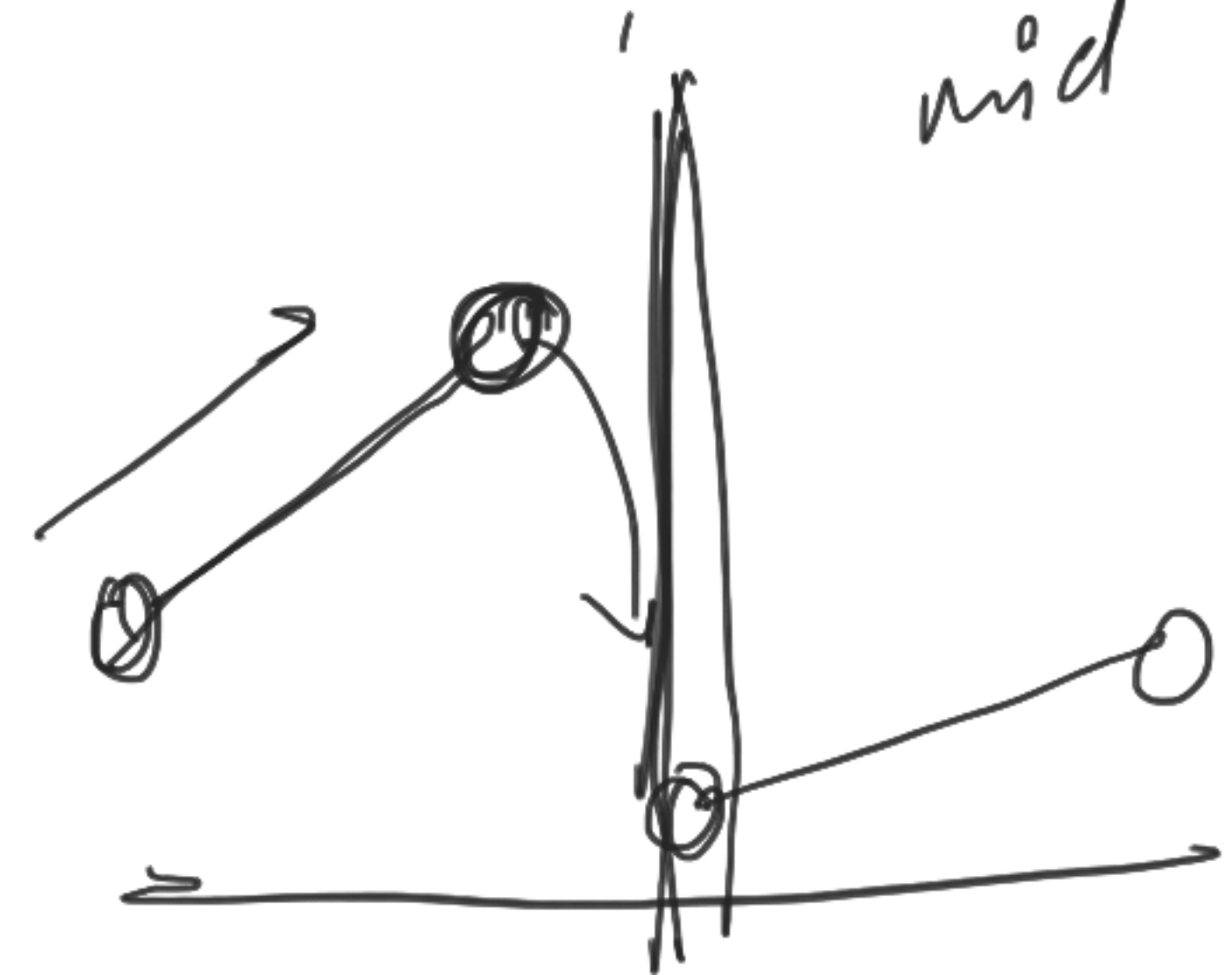
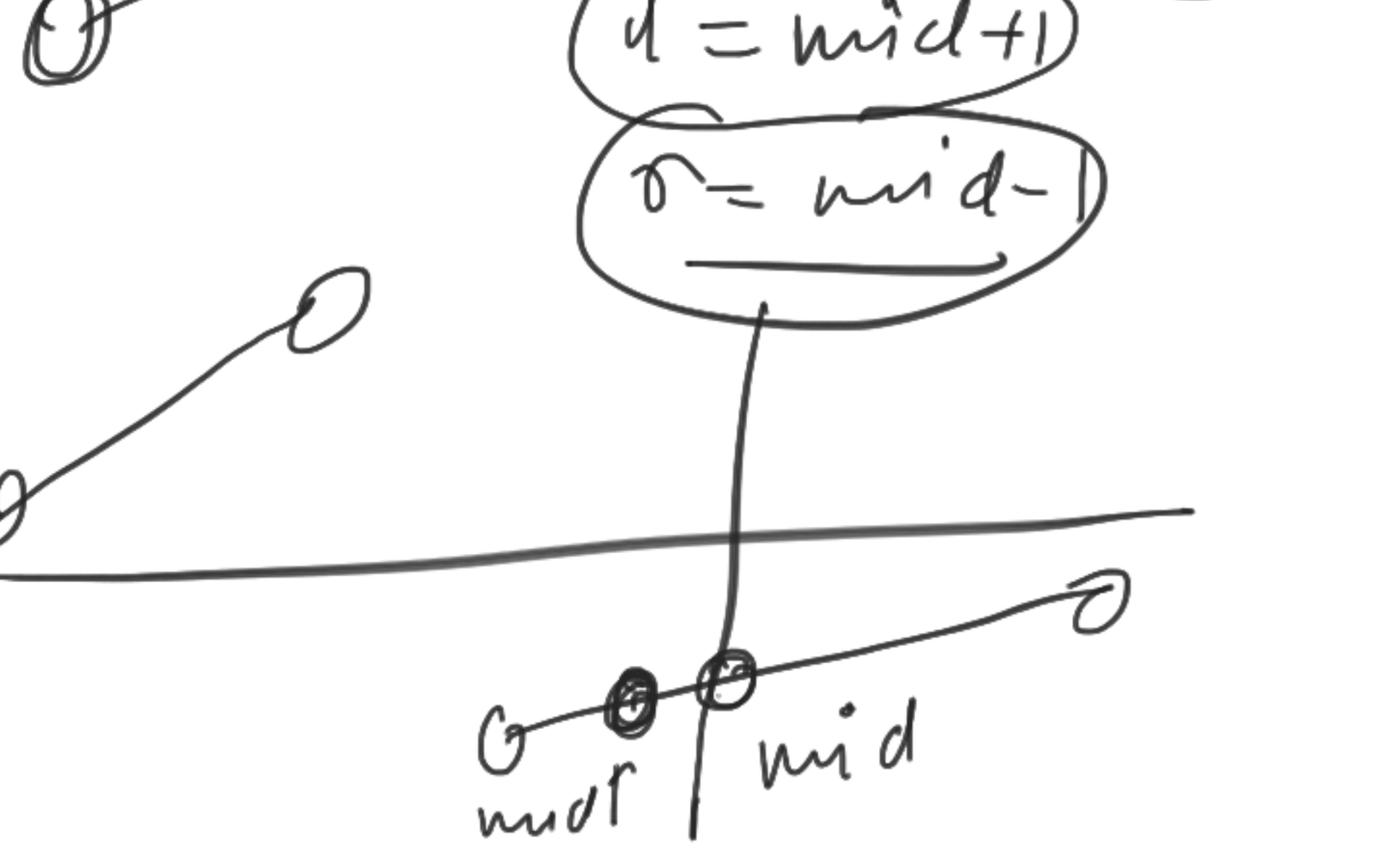
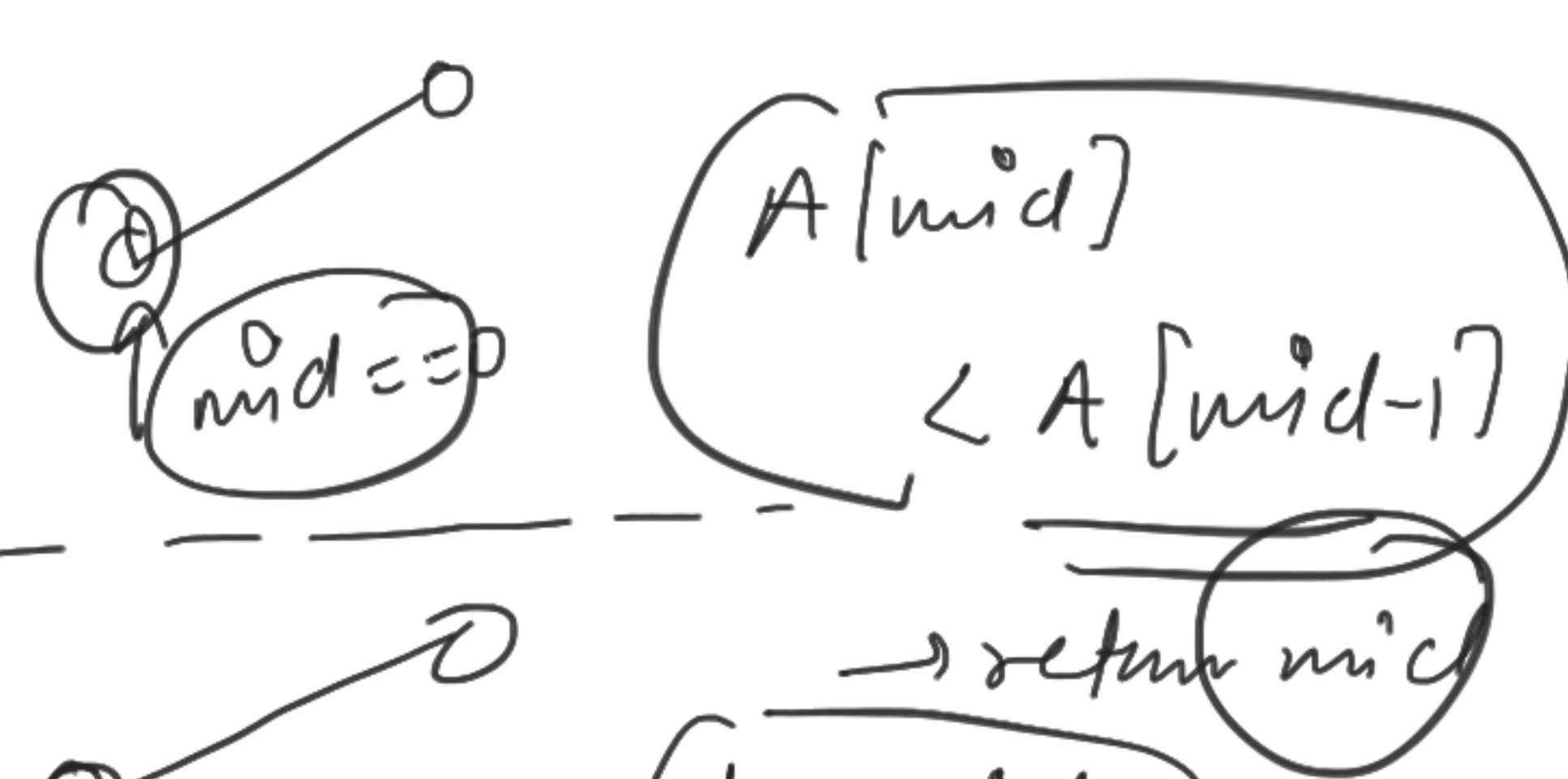
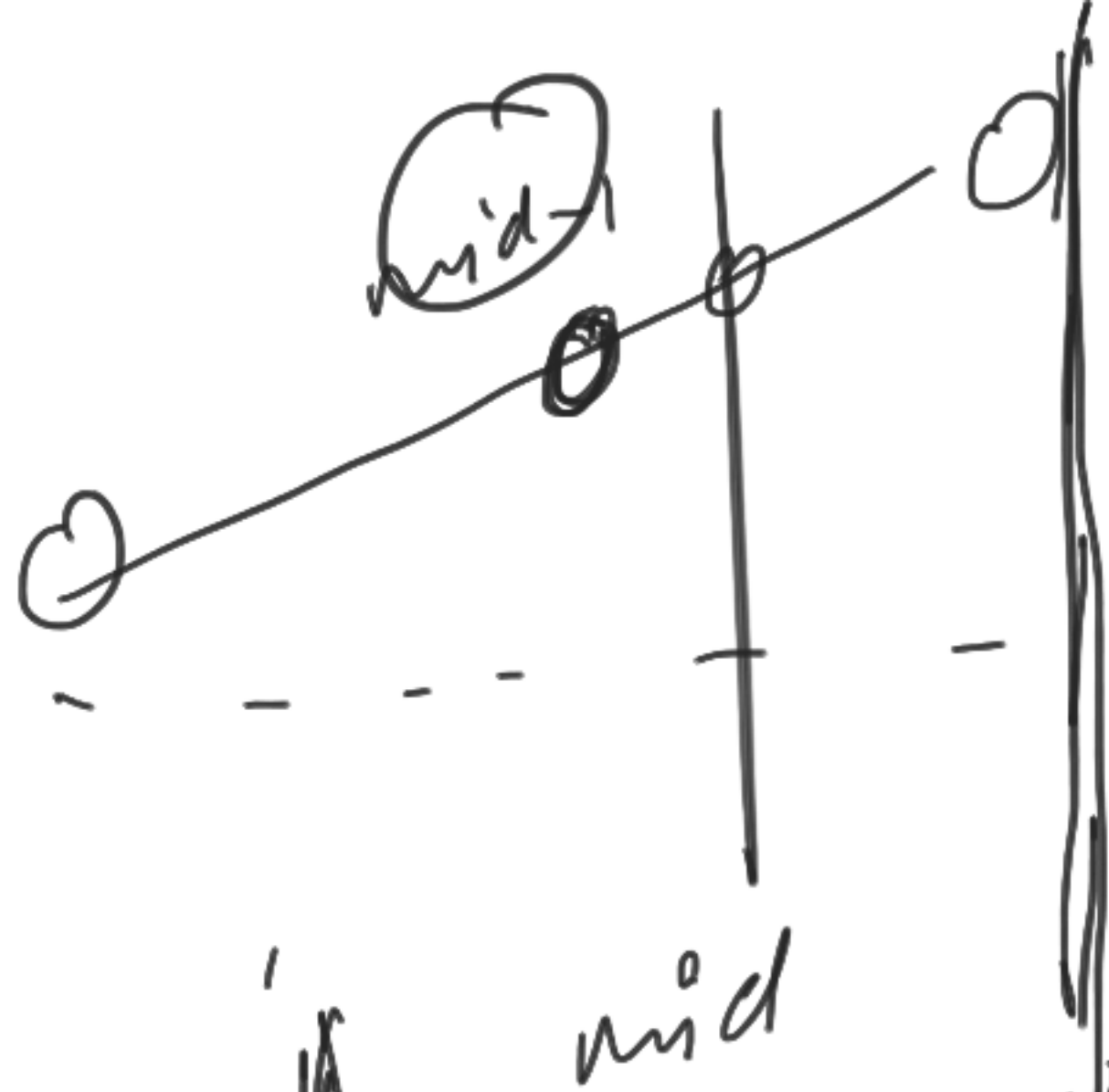
else  $r = mid$

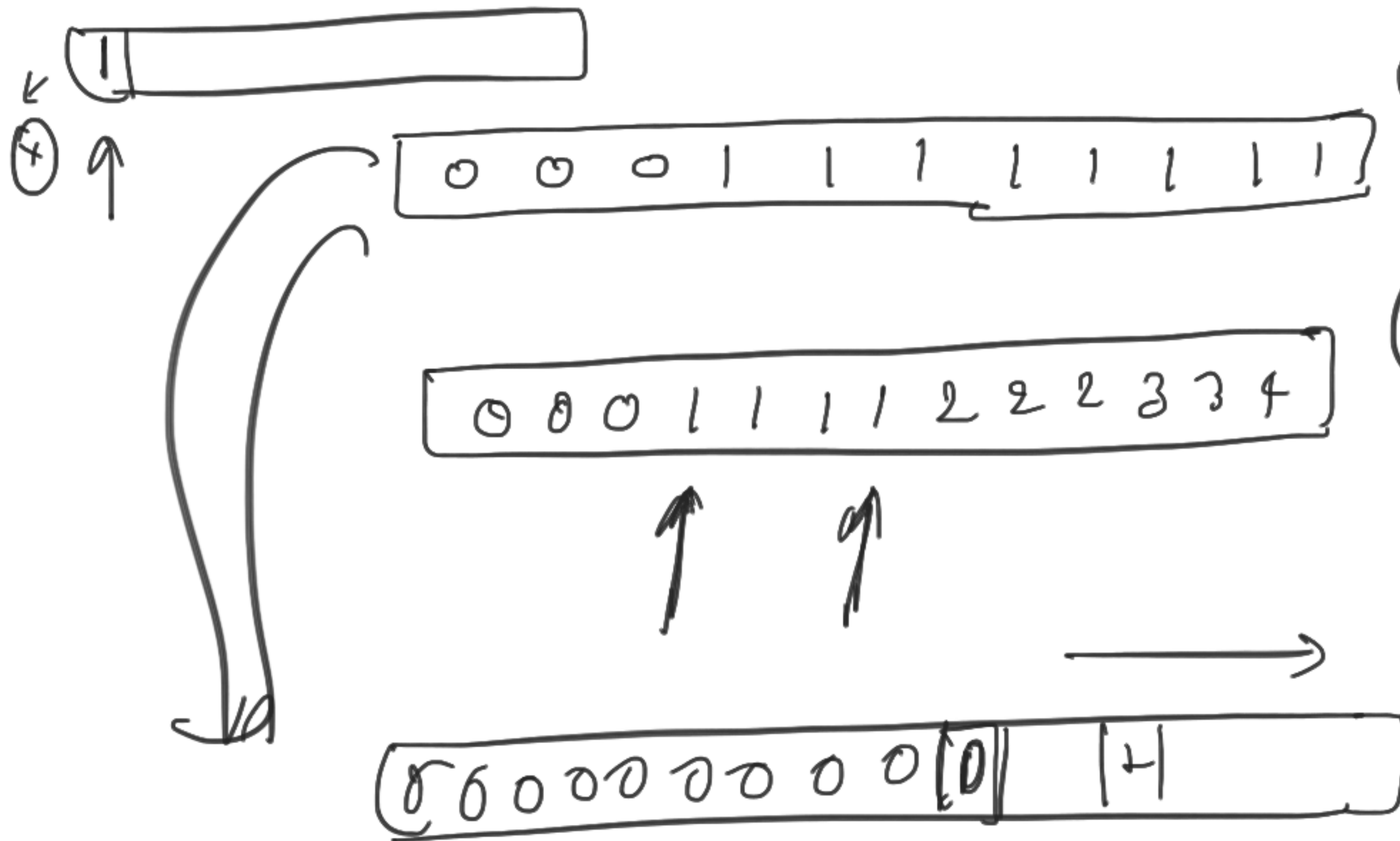
}

return  $A[l]$  or  $A[r]$

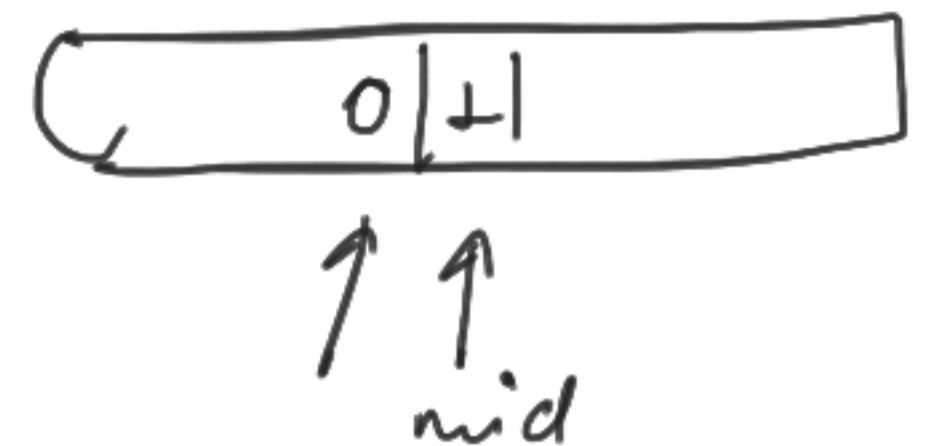




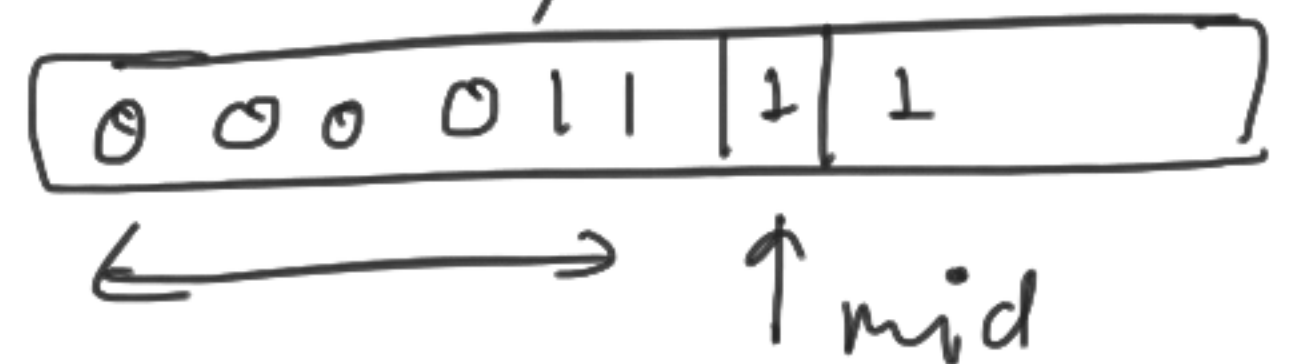




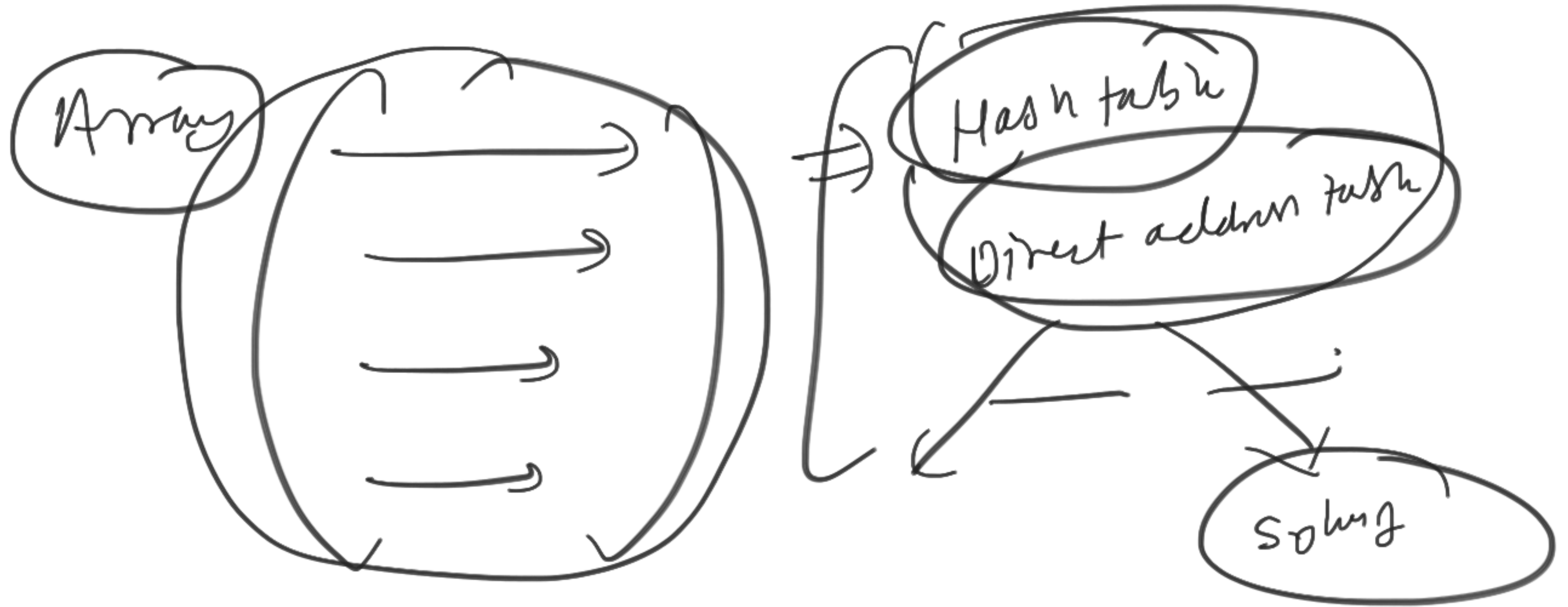
$A[mid] == 1$   
 $\Delta A[mid-1] = 0$   
 target  
 reb mid



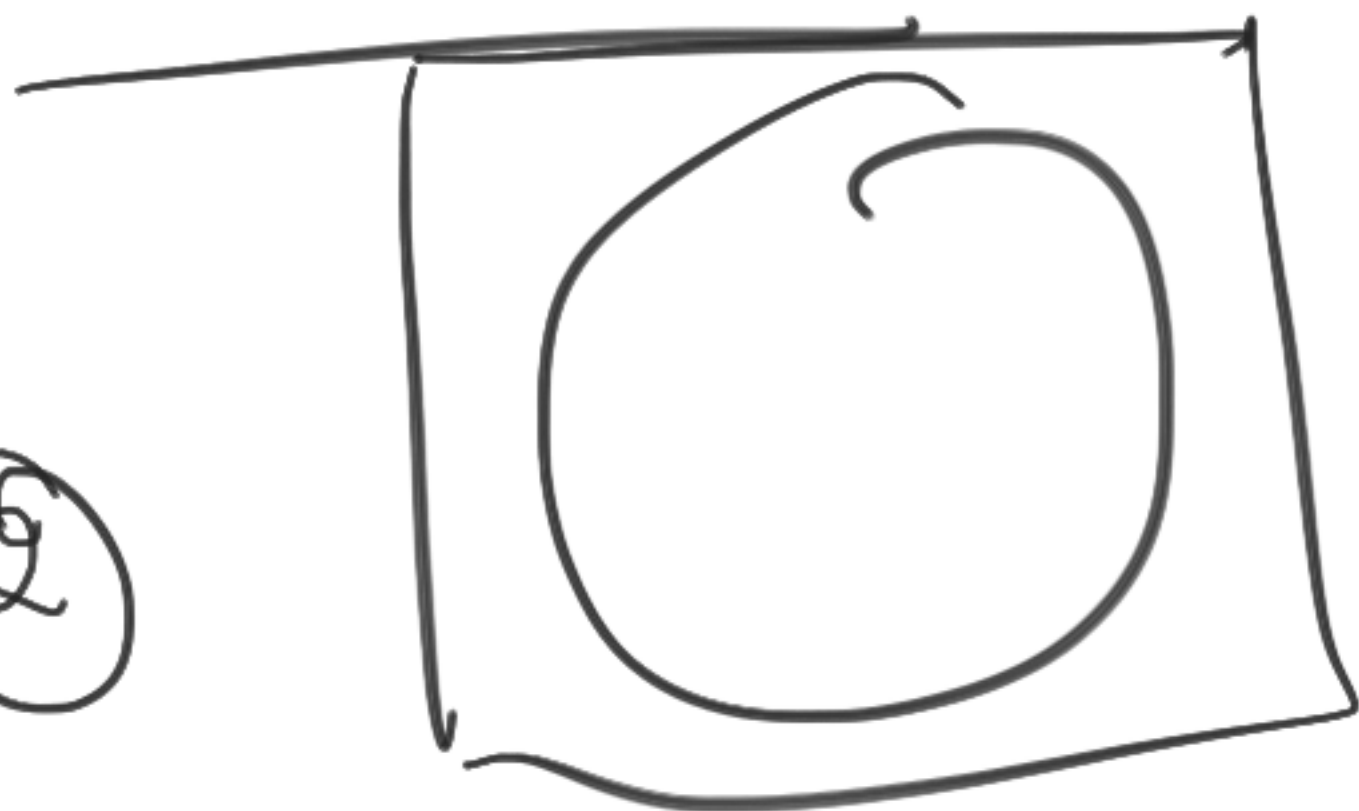
$mid = 0.11$   $A[mid] = 1$   
 $\Delta A[mid-1] = 0$



$(mid == 0 \parallel A[mid-1] == 0) \&\& A[mid] == 1$   
return mid



2



6 m



note

week



0 => simple

