

Intro to Sorting

sort() ← Usage:

C ↓ <u>qsort(a, size...)</u>	C++ <u>sort(v.begin, v.end)</u>	Java <u>Collections.sort()</u>	Python <u>l.sort()</u>
------------------------------------	------------------------------------	-----------------------------------	---------------------------

⇒ $O(N \log N)$

Q.1 Noble Integers

Given array, check if there exists an integer P s.t. (no. of elements greater than P) = P

eg. $A = [3, 2, 1, 3] \Rightarrow \text{True}$

$A = [0, 1, 3, 1] \Rightarrow \text{True}$

1, 2, 4, 5, 6, 8, 10

BRST
Order
/ not
3
2
1
3

3, 2, 1, 3 → ~~3, 1, 3~~

[1, 3, 1, 1]

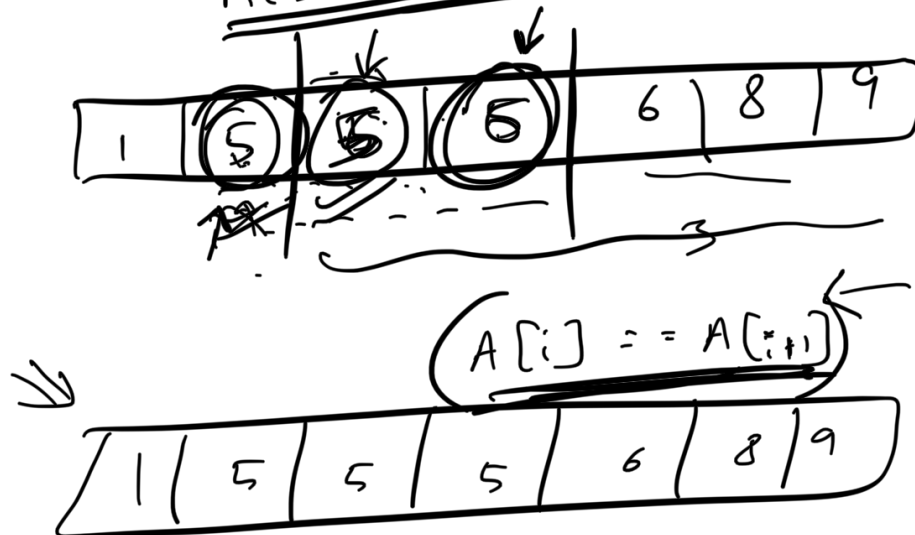
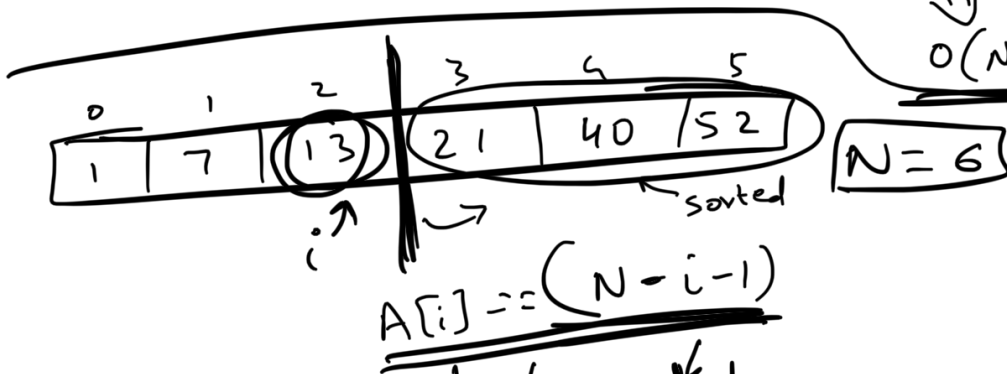
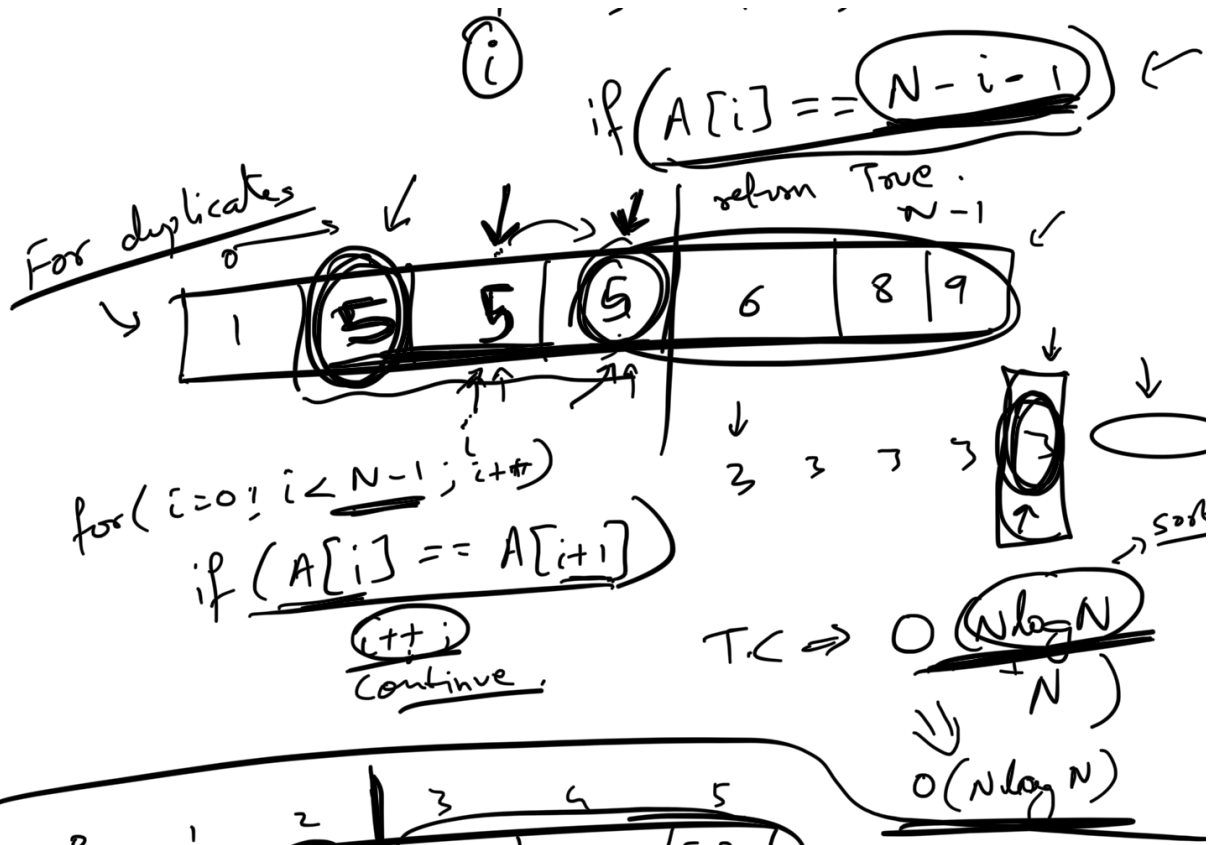
if elements are distinct

1	1	1	3
0	1	2	3

6, 6, 9, 10

1	2	4	6	6	9	10
0	1	2	3	4	5	6

N =



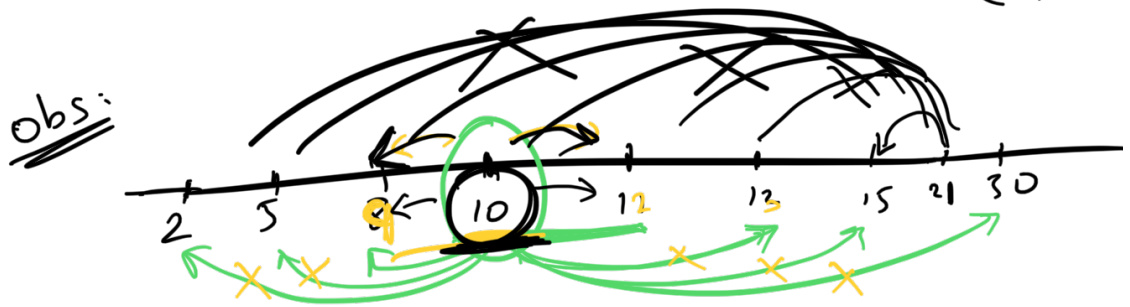
Q.2 Given array find pair of nos. with abs min. difference

e.g. [10, 30, 21, 28, 11, 15]

$\Rightarrow (10, 11)$

(11-10) - - - - -

$[10, 11, 15, 21, \underline{28}, \underline{30}] \Rightarrow (2)$



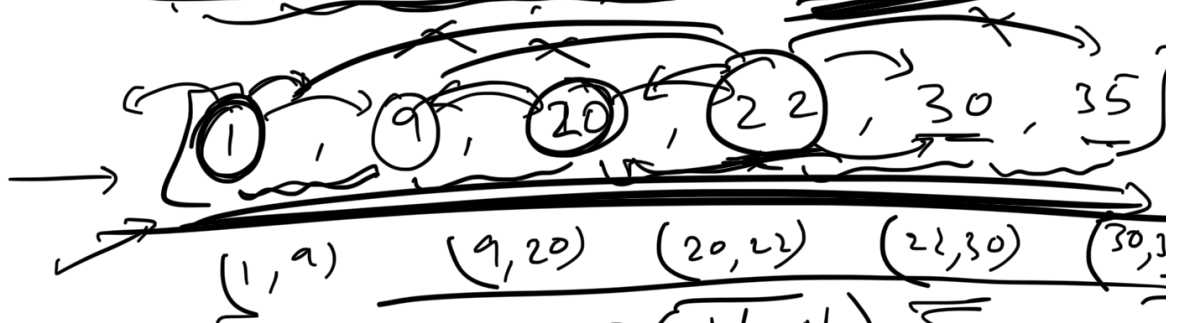
$[10, \textcircled{9}, 5, 6, \textcircled{9}, 8]$

$[8, 19, \underline{25}, \underline{26}, 30, 36]$

Brute force



$O(n^2)$



Modify

T.C $\Rightarrow O(N \log N)$

(Return all pairs in sorted order with min diff)

$[1, 3, 2, 5, 8]$

$\rightarrow [1, 2], [2, 3]$

5 1 2 1

eg $A = [1, 3, 2, 5, 8]$

```

A.sort()
Min = INT_MAX A[1] - A[0]
for (i = 0; i < n-1; i++)
{
    if (A[i+1] - A[i] < Min)
        Min = A[i+1] - A[i]
}
vector<int> ans;
for (i = 0; i < n-1; i++)
{
    if (A[i+1] - A[i] == Min)
        ans.insert(ans.begin(), A[i], A[i+1])
}
return ans;
    
```

$\Rightarrow O(N \log N)$

Q.3 Given array, check if we can form an A.P. using all the values of the array.

(2, 4, 6, 8, 10, ...)

A.P : $(a, a+d, a+2d, a+3d, \dots)$

eg. $[3, 5, 1] \Rightarrow \underline{\underline{\text{True}}}$

$[1, 3, 5]$

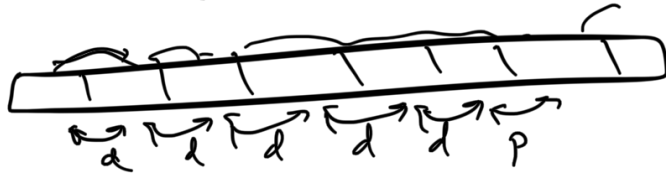
$d=0$ (2, 2, 2, 2, 2, ...)

$d=-3$ (10, 7, 4, 1, -2, ...)

1 - 5 12 8 13, 18, ...

$a \rightarrow (\dots)$

Sort $\rightarrow d \rightarrow a[i] - a[0]$



$(1, 7, 5, 3, 4)$

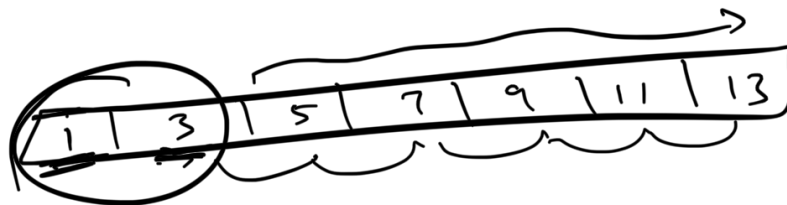
$\times (1, 3, 4, 5, 7)$ T.C $\Rightarrow O(N \log N)$

⑫ $\swarrow [1, 4, 7] = \text{True}$

⑫ $\nwarrow [1, 5, 6] \Rightarrow \text{False}$

Optimisation $\rightarrow O(N)$

$1, 2$
 \downarrow
 $①, 2, 4, ⑥$
 $①, 3, 3, ⑥$



$N =$

Min $\rightarrow O(N)$
 2nd Min $\rightarrow O(N)$
 $O(N)$

Diagram: $1 \mid 7 \mid 9 \mid 11 \mid 5 \mid ③ \mid ① \mid 13$

Annotations: $10(\cdot) d$, Hash map, Hash set, N , $N=7$



min $\rightarrow ①$
 2nd min $\rightarrow ③$

~~AP~~
 $1, 3, 5, 7, 9, 11, ⑬$
 $\Rightarrow (1, 3, 5, 7, 9)$
 \times

$[5, 5, 5, 5]$ \downarrow (5)
 $\frac{n}{2} [\min + \max]$ $\swarrow \searrow$ $[1, 2, 4, 6]$ ~~4~~
 $\searrow \swarrow$ $[1, 3, 3, 6]$ ~~X~~

$\frac{n}{2} [\min + \max]$ $\swarrow \searrow$ $[2, 4, 6, 8]$
 $\searrow \swarrow$ ~~$[2, 5, 5, 8]$~~

$TC \Rightarrow O(N \log N)$ $S.C \Rightarrow O(1) =$
 $TC \Rightarrow O(N)$ $S.C \Rightarrow O(N)$ ~~2~~

$\Rightarrow [2, 4, 6, 8, 10]$
 ~~$[2, 4, 7, 7, 10]$~~

Q.4 Given array of colors $\rightarrow [0, 1, 2]$,
sort the array.

E.B
ms $\Rightarrow [1, 0, 2, 2, 1, 0, 0, 2, 1] \leftarrow O(N)$
 \Downarrow $[0, 0, 0, 1, 1, 1, 2, 2, 2]$

2-pointer
 count
 count
 count

0	= 3
1	= 3
2	= 3

 $O(N)$
 $O(N)$

$\{0, 0, 0, 1, 1, 1, 2, 2, 4\}$
T.C $\Rightarrow O(N)$
S.C $\Rightarrow O(1)$

Q.5 Given array of distinct elements, return K^{th} min. element.

eg. $[2, 0, 5, 8, 3, 1]$ $K=3$
 \rightarrow sort 2 $K=4$

sort & return $a[K-1] \Rightarrow O(N \log N)$

\Downarrow modification

Total swaps $\leq K$

$[2, 0, 5, 8, 3, 1]$

$\rightarrow [0, 2, 5, 8, 3, 1]$
 $\rightarrow [0, 1, 5, 8, 3, 2]$
 $\rightarrow [0, 1, 2, 8, 3, 5]$

$\left\{ \begin{array}{l} [2, 1, 5, 8, 3, 0] \\ [0, 1, 2, 3, 5, 6] \end{array} \right\}$

$K=3$

$K < \log N$

T.C \Rightarrow $O(K \cdot N)$

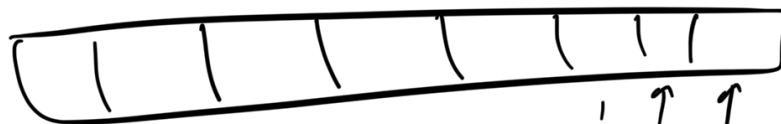
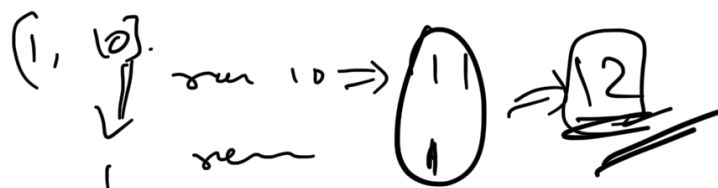
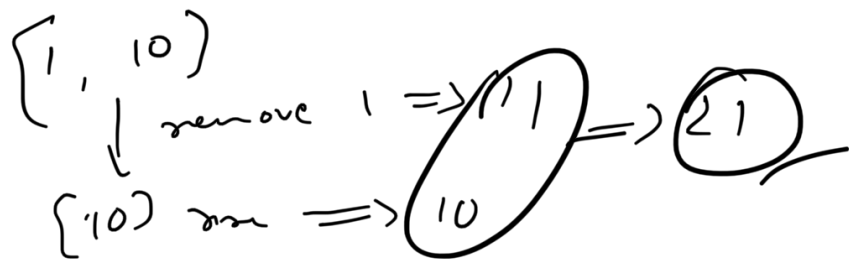
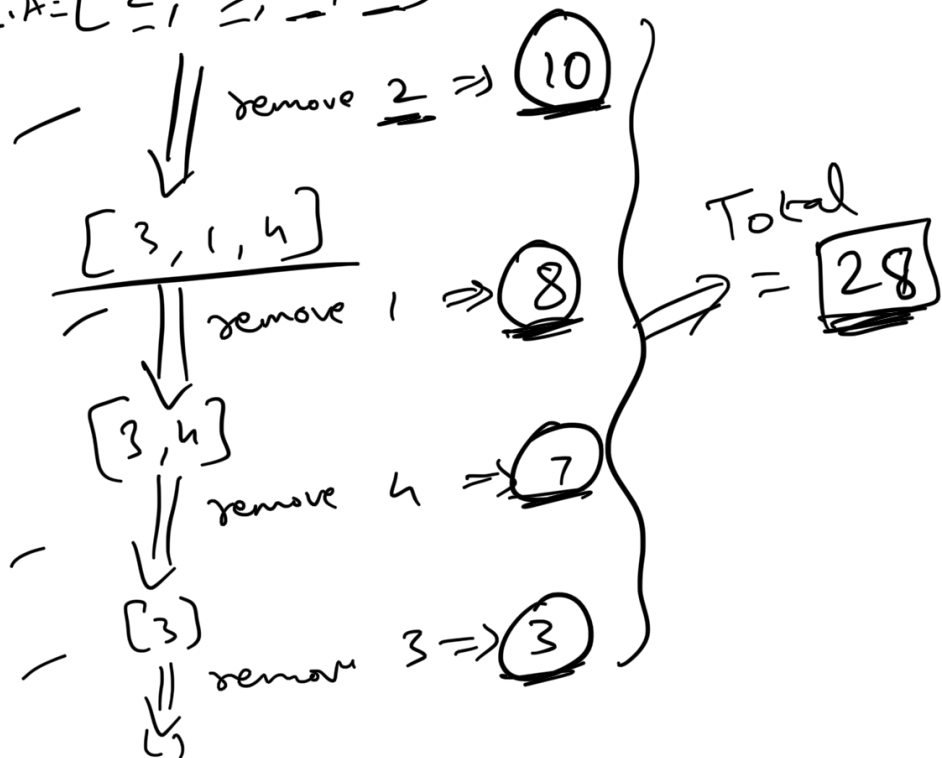
$K=N$

Sort the whole array \Rightarrow $O(N^2)$
 (collection)

Sort


Q.6 Given array, remove all elements.
 Cost of removing an element = $\frac{\text{Sum of elements before we remove this element.}}{2}$

Minimize Total cost.
 eg. A = [2, 3, 1, 4]



obs:-

[2, 4, 9, 15]
 1st step
 Total \Rightarrow 30

[2, 4, 9] \Rightarrow 

[2, 4]

\Downarrow
 (2)
 \Downarrow

(Greedy Approach)

Sort & start picking element from last.

[2 | 7 | 15 | 20 | 30]

$O(N \log N)$

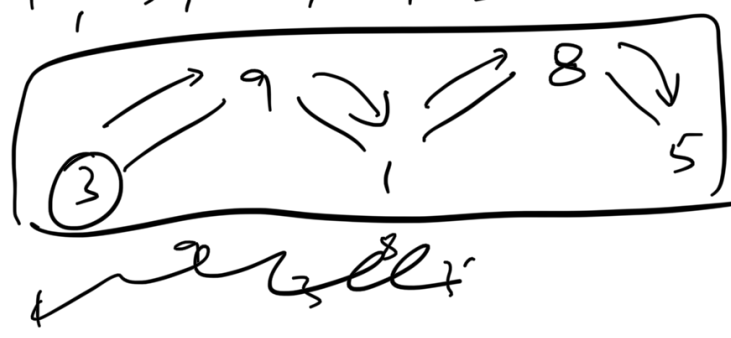
total = 30 - 20

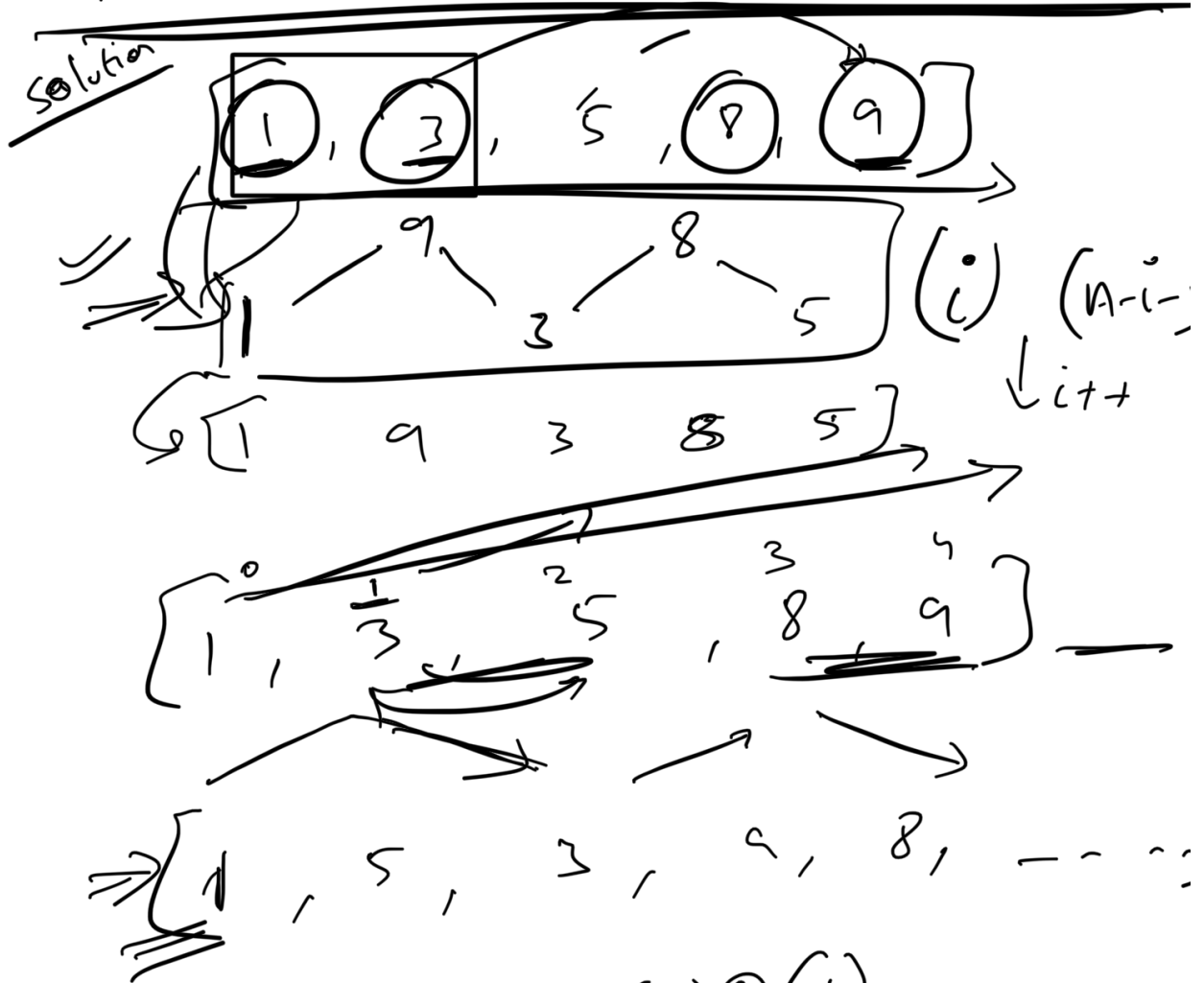
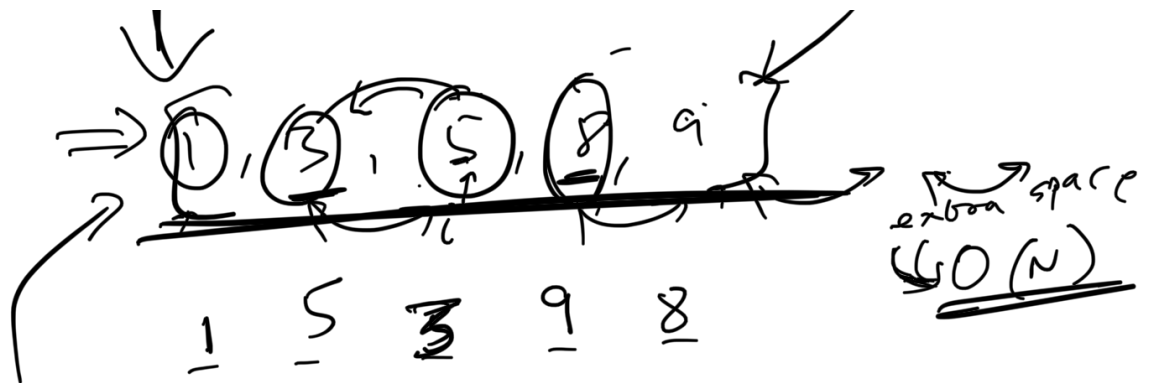
$O(N)$

(of distinct integers)

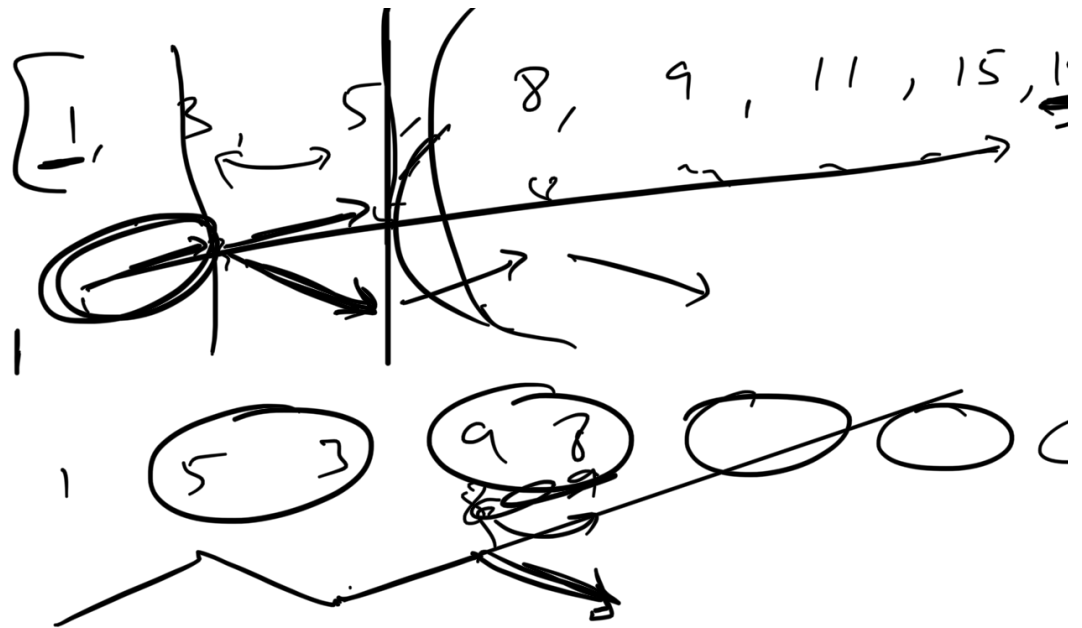
Q.7 Given array, arrange the array in wave form



1 possible \rightarrow [1, 3, 9, 8, 5]
 
 result



S.C $\Rightarrow O(1)$
 T.C $\Rightarrow O(N \log N)$
Sort(a, b, n)
 for (i=1; i<n; i+=2)
 {
 swap(a[i], a[i+1])
 }
 return a;



Q.2 examples

