

Sorting

Date: ____/____/____

Highest element will reach end first

Bubble
Selection
Insertion
Merge
Count.

Bubble Sort:

7 9 6 12 10 29 5 3

n=8
first pass
n-1=7

1st pass
2nd pass

7 6 9 12 10 5 3 29 → first high elem reached end.

12 29
10 12 29

void bubblesort (int a[], int n) {

for (int i=0; i<n; i++) {

for (int j=0; j<n-i; j++) {

if (a[j] > a[j+1]) {

int t = a[j];

a[j] = a[j+1];

a[j+1] = t;

}

→ Here n-i because in every pass

one highest elem going to end

It's not necessary compare highest elements that reached end

→ 8-0-1 →

Best Case: Already Sorted, O(1)



Selection Sort: Find Smallest element - and place it at 0th pos.
 2) find largest element and place it at n-1, ...

20 24 4 16 5 2 31 23
 → 20 24 4 16 5 20 31 23

```

for (int i=0; i<n; i++)
  for (int j=i+1; j<n; j++)
    if (a[i] > a[j])

```

Swap code

```

void Selection (int a[], int n) {

```

```

  for (int i=0; i<n; i++) {

```

```

    int ele = a[i], idx = i;

```

```

    for (j=i+1; j<n; j++) {

```

```

      if (a[j] < ele) {

```

```

        ele = a[j]

```

```

        idx = j;
      }
    }

```

```

    Swap (a[i], a[idx])
  }
}

```

Worst Case
 n^2



Date: ____/____/____

Insertion Sort:

	0	1	2	3	4	5
arr		3	12	7	5	16
j		i				

ele = 3 = i Compare j & ele arr[j]

j = -1 → Stop

3	20	12	7	5	16
---	----	----	---	---	----

j > i

ele = i = 12

3	20	20	7	5	16
---	----	----	---	---	----

My
Steps

```
void insertion (int arr, int n) {
```

```
for (int i = 1; i < n; i++)
```

```
    ele = arr[i]
```

```
    for (int j = i - 1; j >= 0; j--)
```

```
        if (arr[j] > ele)
```

```
            arr[j+1] = arr[j]
```

BC	$O(n)$
W.C	$O(n^2 + 1)$



```
void insere (int arr, int n)
```

```
for (int i = 1; i < n; i++) {
```

```
    int ele = arr[i], j = i - 1;
```

```
    while (arr[j] > ele && j >= 0) {
```

```
        arr[j+1] = arr[j]
```

```
        j--;
```

```
    }
```

```
    arr[j+1] = ele;
```



→ n

- n+1
n(n+1)
n²+1

Elections

Candidate = 10

People = 102

→ given

votes to candidates / any one of the candidate

a

2	3	4	3	3	1	1	7	4	9	8	10 * N
											TC

Cell contains Candidates

int votes = 0, count = 0;

```
for (int i = 0; i < n; i++) {  
    (int j = 0; j < 10; j++)  
    if (a[i] == j)
```

→ using loop

```
int elec (int a[], int n) {
```

```
    int winner = 1, vote = 0;
```

```
    for (i = 0; i <= 10; i++) {
```

```
        int count = 0;
```

```
        for (j = 0; j < n; j++) {
```

```
            if (a[j] == i) {
```

```
                count++
```

```
            }
```

```
            if (count > votes) {
```

```
                votes = count;
```

```
                winner = i;
```

```
            }  
            return winner;
```

TC	SC
10 * N	1

N * m
↓
worst
Candidate



Date : / /

votes 17 =

2	3	9	3	7	1	1	7	3	4	3	3	2
---	---	---	---	---	---	---	---	---	---	---	---	---

Count [] =

0	1	2	3	4	5	6	7	8	9	10
//	2	2	5	1	0	0	2	0	1	0

Count array contains respective votes $\text{count}[i] = 1^{\text{st}} \text{ candidate votes}$ total
 $\text{int count}[n] = \text{new int}[n+1];$ $[2] = 2^{\text{nd}}$ " "

```
int count[] = new int[n+1];
for (int i=0; i<n; i++) {
```

$$\text{Count}[a[i]] + 1$$

२

```
int max = a[0], win = 1;
```

```
for (int i=1; i<=10; i++) {
```

```
if (count[i] > max) {
```

max = Count[i]

$$WIM \leq P$$

return wins

TC	SC
N+10	11

Q: Sorting voter array using Count array

Count array

Date: _____

int arr[] = new int[10];
 for (int i = 1; i <= 10; i++) {
 // count array size - Not candidate

if (count(i) > 0) {

for (int j = 0; j < count(i); j++) {

~~arr[i] = count[i];~~
 votes[k++] = i;

for (int i : votes)
 print(i)

TC: $N + M$

SC: M

to no. of candidates

because we consider count of size of candidate

① $R \geq b - a + 1$

② arr[R]

③ $cnt[arr[i] - a]++$

Count array fails at large values

Build your dreams

Q. Ans: 3, 5, 9, 16, 21, 40

Date: / /

Bm: 7 12 20 21 50

df: 3 5 7 9 12 16 20 21 21 40 50

~~void print(int a[], int n, int B[], int m){~~

My thought
Using
2-pointe

```
for (int i = 0; i < n; i++) {
```

```
for (int j=0; j<m; j++) {
```

```
if (a[i] > a[j]) {
    swap(a[i], a[j]);
}
```

$$\rho_{\text{eff}}(a_{\text{eff}})$$

Working only
on sorted array

Have same
arr

Technique
8. work as if arrays are
sorted
Bubble force.

$$T.C. : N + M + \frac{(N+M)^2}{2} + (N+M)$$

$$S.C. : N + M$$

2) C_{sum} : N + (N+M) + (N+M), N+1
 ↑ ↓ ↓
Copy Adto iteration print array
C[] step/set

Using 2-pointer technique works only on sorted array of p's.

```
3) void printArray (int a[], int n, int B[], int m) {
```

for Cinf

```
int i=0, j=0;
while (i < m && j < n) {
```

if $(a[i] > a[j])$ &

```
print(alist);
```

$$j++;$$

```
else print(a[i])
```

$$1 + 1; 3$$

```
while (i < n) { print A[i+1] }
```

```
while (j < m) { print A[j++]}
```

Ans

TC: $N+M$
SC: 1



Print in Sorted order into Single Array 'A'
 Add B to A in Sorted order.

Date: ____/____/____

Q: A: 2 4 12 14 20
 B: 5 13 16 21 40

Solution:

1) $M + N$, 1
 ↑ BS/SS/2 SNA
 Copy B to A

2) $N * (N - M)$, 1
 ↑ Preservation

3) $N + N$, N
 2-pointer
 Copy C to A
 A & B → C

4) $M + (N - M)$, 1
 2 pointer

Using 2-pointer (i, j) - from back side placing max elements at end and continuing.

Date: ____/____/____

int i = n-m-1, j = m-1, k = n-1;

if (A[i] > B[j]) {

 A[k] = A[i]

 i--;

 k--;

}

else {

 A[k] = B[j]

 j--;

 k--;

while (j >= 0) {

 A[k--] = B[j--];

}

→ why only j is

if elements
which are
not traversed

still in

arr only

INFRA DEVELOPERS PRIVATE LIMITED

Build your dreams



Q: 0 1 1 0 1 0 1 1 1 0 0 1
Sort 0's and 1's

Date: ____/____/____

1) $N^2, 1$ (BS/SS/SS)

2) $\frac{N}{2} + \frac{(N+2)}{2}$, $\frac{2}{2} \rightarrow$ Count array size
Count array Count sort

3) $O(N)$ \rightarrow to-do

Using 2 pointers?
 $i=0, j=n-1$

```
while (i < j) {  
    if (a[i] == 0)  
        i++
```

```
    else if (a[j] == 1)
```

```
        j--
```

```
    else if (a[i] > a[j])
```

```
        // Swap
```

```
        i++
```

```
        j--
```

```
}
```

INFRA DEVELOPERS PRIVATE LIMITED

Build your dreams



0 1 2 3 4 5 6 Date: 8 / /

a: 3 10 12 -5 17 18 5 30 25

rules:

- 1) $i < j$
- 2) $a[i] > a[j]$

dp: How many pairs can obtain following above rules is.

dp: 8

Sol:
Case-1!

int Count = 0;

for (int i = 0; i < n; i++) {

for (int j = i+1; j < n; j++) {

if ($i < j$ & $a[i] > a[j]$) {

Count++;

}
return Count;

TC: N^2 SC: 1

INFRA DEVELOPERS PRIVATE LIMITED

Build your dreams

$0 \leq \text{pair} \leq n(n-1)/2$

min

max

↓

↓

for

for

ascending
ordered
array

descending
sorted
array


```
void ms (int a[], int l, int h) {
    if (low == high) return;
    int mid = (l+h)/2;
```

```
    ms (a, l, mid)
```

```
    ms (a, mid+1, h)
```

```
    Merge (a, mid, h)
```

```
}
```

```
void merge (int a[], int low, int mid, int hi) {
```

```
    int P1 = low, P2 = mid+1, k = 0;
```

```
    int temp[];
```

```
    while (P1 ≤ mid && P2 ≤ high) {
```

```
        if (a[P1] ≤ a[P2])
```

```
            temp[k++] = a[P1++];
```

```
    else
```

```
        count++;
```

```
        temp[k++] = a[P2++];
```

```
    while (P1 ≤ mid) { temp[k++] = a[P1++] };
```

```
    while (P2 ≤ high) { temp[k++] = a[P2++] };
```




Merge Sort: Divide and Conquer.

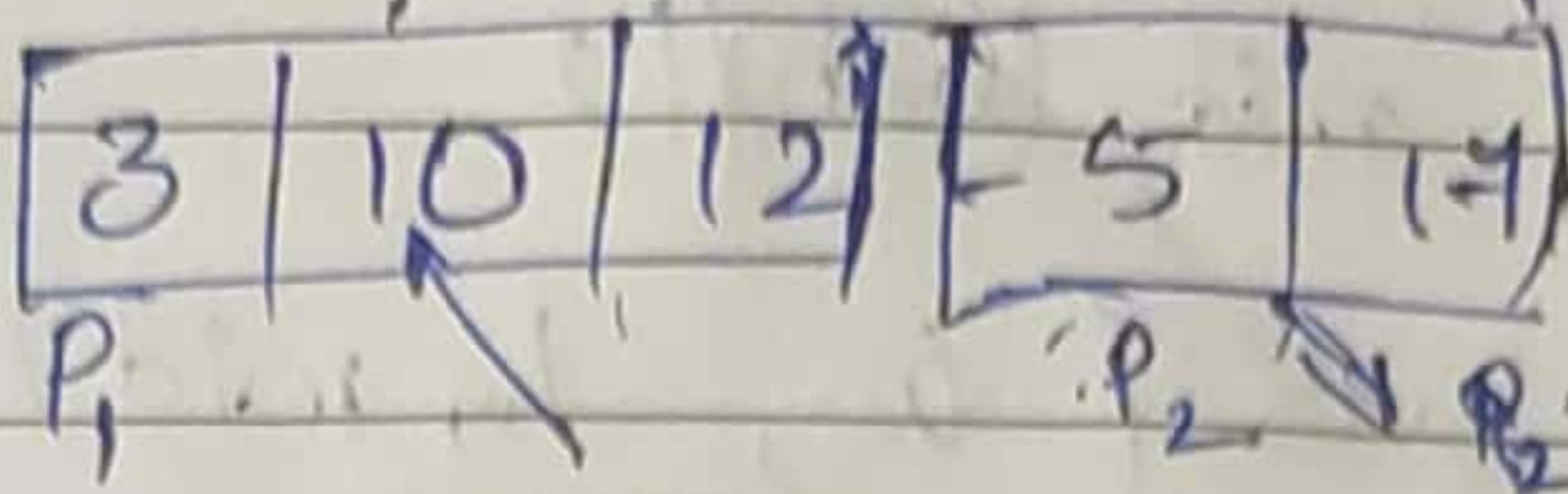
Date: ____/____/____

0 1 2 3 4 5 6 7 8
a: 3 10 12 5 17 18 5 30 25

1st half.

$$\frac{0+8}{2} = 4$$

→



$P_1=0, P_2=0, K=0 \dots W/2$

$P_1 < P_2$

$a[K] = a[P_1]$

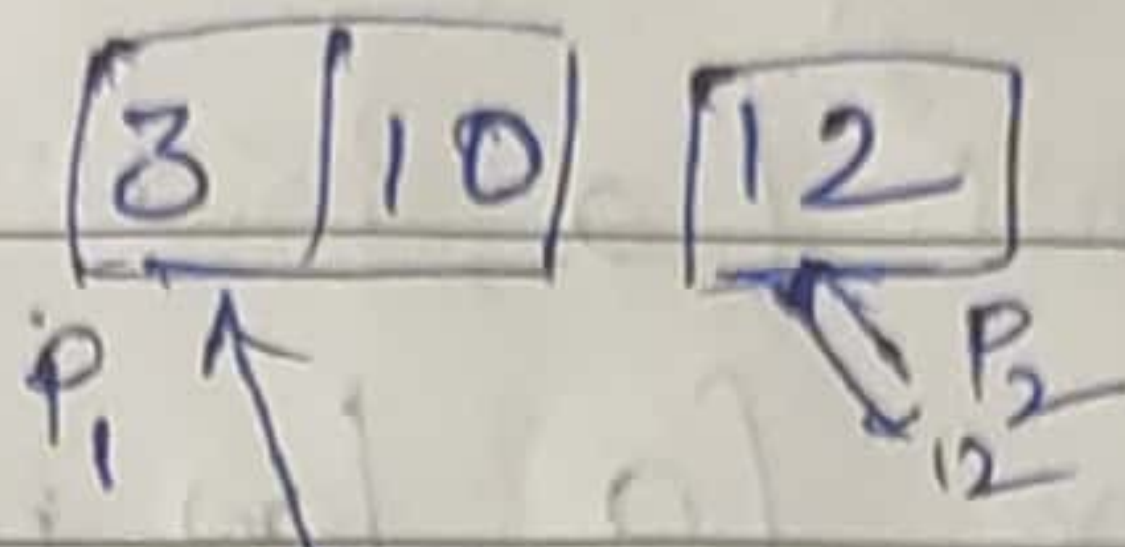
else P_1++
 P_2++

→ Compare using

2 ptr's: P_1, P_2

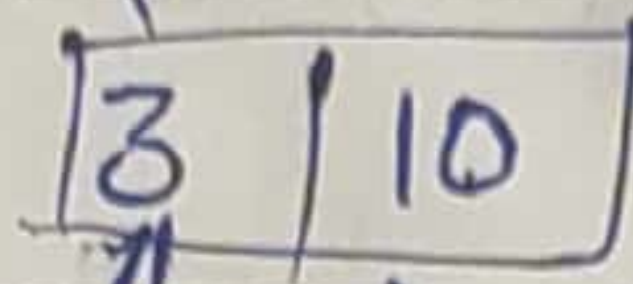
$$\frac{0+4}{2} = 2$$

→



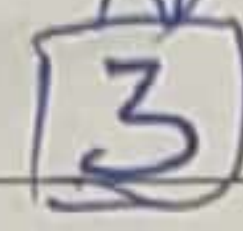
$$\frac{0+2}{2} = 1$$

→



$$\frac{0+1}{2} = 0$$

→



- Compare

Second half

$P_1 < P_2$ $a[K] = a[P_1]$

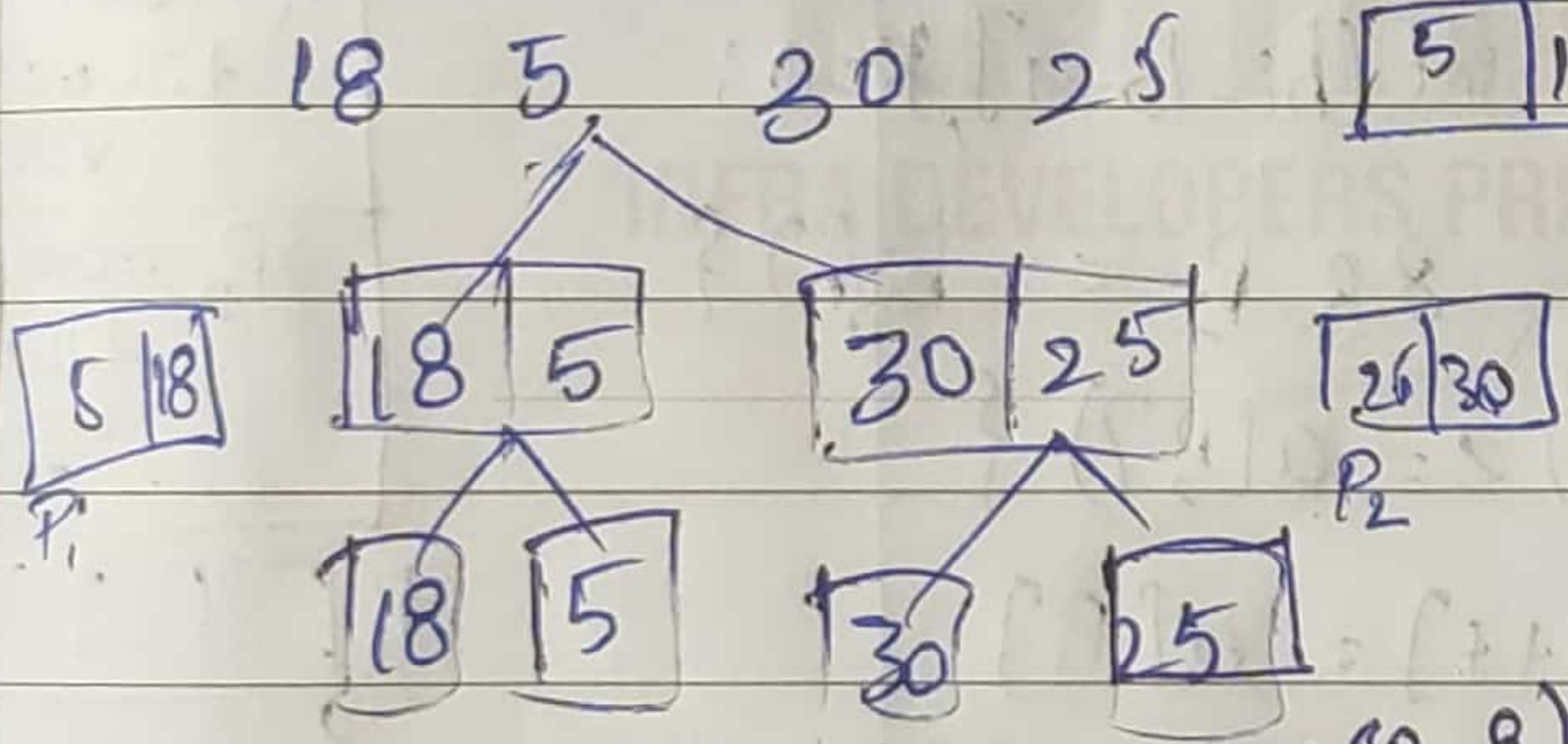


$$m = (l+h)/2$$

$ms(a, l, m)$

$ms(a, m+1, h)$

$merge(a, l, m, h)$



$m(0, 8)$

$ms(0, 4)$

$ms(5, 8)$

$merge(0, 4, 8)$

$ms(0, 2)$

$ms(3, 4)$

$merge(0, 2, 4)$

$ms(5, 6)$

$m(7, 8)$

$merge(5, 6, 8)$

$ms(0, 1)$

$ms(2, 3)$

$ms(3, 3)$

$ms(4, 4)$

$m(3, 3, 4)$

$ms(5, 5)$

$ms(6, 6)$

$ms(7, 7)$

$ms(8, 8)$

$merge(7, 7, 8)$

$merge(0, 1, 2)$

$merge(2, 3, 4)$

$merge(5, 6, 8)$

$ms(0, 0)$

$ms(7, 7)$

$merge(0, 7, 8)$

$ms(0, 0)$

$ms(7, 7)$

$merge(0, 7, 8)$



$$\frac{0+8}{2} = 4$$

$$0 - 4$$

$$\text{low} - \text{mid}$$

$$5 - 8$$

$$\text{mid} - \text{high}$$

→ to merge in sorted order

void ms (int a[], int low, int high) {

if (low < high) return;

int mid = (low + high) / 2;

ms (a, low, mid);

ms (a, mid+1, high);

} Merge (a, low, mid, high);

0, 4, 8

→ T(N/2)

→ left

→ T(N/2)

→ right

→ T(N)

[a, b] base

temp array size = high - low + 1

pointers $P_1 = \text{low}$, $P_2 = \text{mid} + 1$

void merge (int a[], int low, int mid, int high) {

int $P_1 = \text{low}$, $P_2 = \text{mid} + 1$;

int temp[] = new int [high - low + 1], k = 0;

while ($P_1 \leq \text{mid}$ && $P_2 \leq \text{high}$) {

if ($a[P_1] \leq a[P_2]$) {

temp[k++] = a[P1];

P_1++ ;

}

else

temp[k++] = a[P2];

P_2++ ;

}

}

while ($P_1 \leq \text{mid}$) { temp[k++] = a[P1];

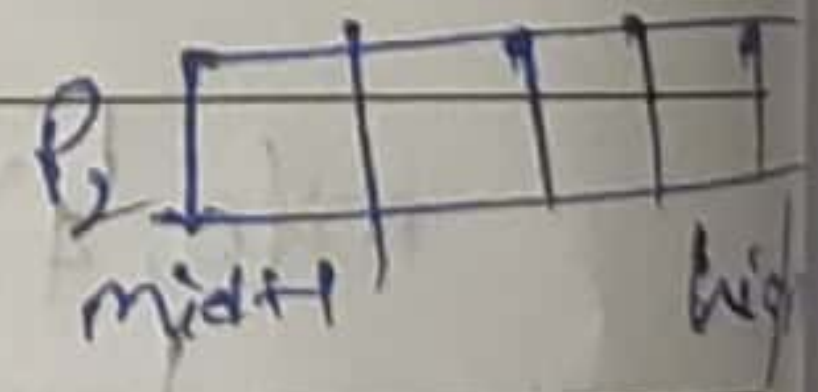
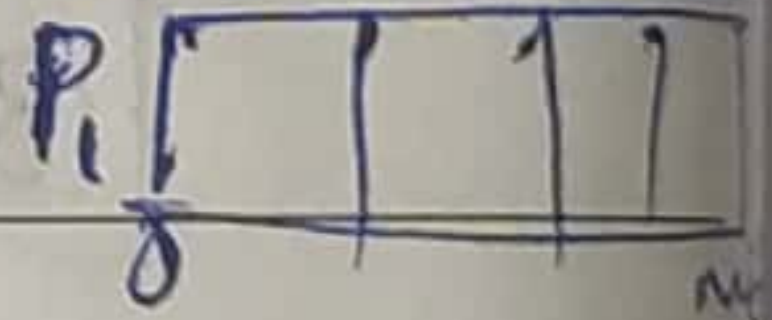
P_1++

while ($P_2 \leq \text{high}$) { temp[k++] = a[P2];

P_2++

P_1 moves till mid value

P_2 moves till end high index





Date: ____/____/____

Copy temp[] to a[] :-

for (int i = 0; i < hi + low + 1; i++) {

 a[i + low] = temp[i];

}

}

TC: $2T(N/2) + N$

$N \log_2 N + N$

K

Sum of pair $= K$ Using

Sort & 2 pointer

Date: ____/____/____

a: -5 12 6 9 20 19 30

Rules 1) $i \neq j$

2) $a[i] + a[j] = K$

Case-1:

for (int i=0; i<n; i++)

for (int j=i+1; j<n; j++)

if ($a[i] + a[j] == K$)

return true

1) N^2 ,

2) $N \log n \neq n$
↓
Sorted

Since Sorted
2 pointer working

Case-2)

while ($P_1 < P_2$) {

if ($(a[P_1] + a[P_2]) == k$)

return true;

else if ($(a[P_1] + a[P_2]) < k$)

P_1++

else ($(a[P_1] + a[P_2]) > k$)

P_2--

}

return false;

K

Linear

Searching

$k=7$

Search:

20

40

60

4

stop here

Date: / /

28

41

for

if
return

Binary Search

Search Space (ordered elements)

a: -5 40 20 36 29 15

K: 29

Step: Sort Array

Sorted a: -5 15 20 29 36 40

Iterative approach

while ($l <= h$) {

int mid = $(l + h) / 2$ = 2

if ($a[mid] == k$) return true

if ($a[mid] < k$) \rightarrow right

$l = mid + 1$ build your dreams

else if ($a[mid] > k$) \rightarrow left

{
h = mid - 1

return false;

}



Date : ____/____/____

Recursive Approach

```
bool binsearch (int a[], int l, int h, int k) {  
    if (l > h) return false;  
    int mid = (l+h)/2;  
    if (a[mid] == k) return true;  
    if (a[mid] > k)  
        return binsearch (a, l, mid-1, k) || -  $T(N/2)$  or  
        binsearch (a, mid+1, h, k) -  $T(N/2)$   
}
```

$$T(N) = T(N/2) + 1 \quad \text{if sorted array}$$

$\lceil \log_2 N \rceil$ \rightarrow sorted array

$$T(N) = n \log n + \log n \quad \text{if unsorted array}$$

\uparrow
merge sort