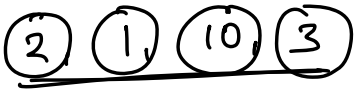
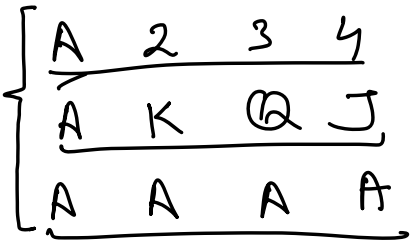
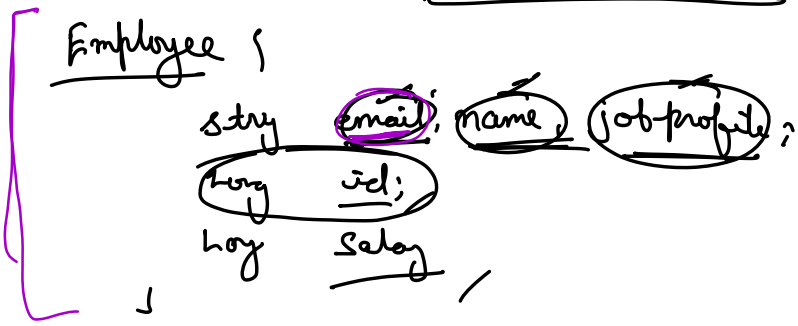
Sorting







Java	C	C++	Python
MergeSort list A Collection.sort(A); int A[] Array.sort(A)	q.sort(---) quickSort	sort(v.begin(), v.end())	list l l.sort()

O(n log n)

(Merge Sort or Quick Sort) ⟹ O(n log n)

Amazon + Goldman Sach

Q. Given an array. Check if there exists a Noble integers.

Noble int: if the no. of int greater than a no. is equal to its val.

3, 2, 1, 3

4, 6, 2, 1, 7, 3 3

1 < 2

1+1

```
for(i=0; i<n; i++){
    cnt=0;
    for(j=0; j<n; j++){
        if(a[j]>a[i])
            cnt++;
    }
    if(cnt==a[i])
        ret true;
}
ret false;
```

a[i]

3, 2, 1, 3

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

3, 3, 2, 1, 0

3, 2, 1, 0

TC: $O(n^2)$

All no. are unique

3, 2, 1, 3

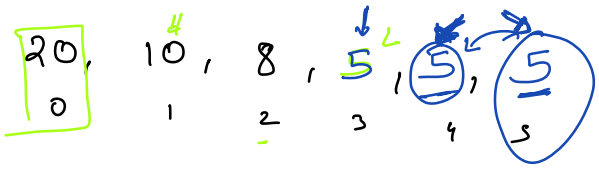
1 2 3 3

0 1 2

20, 10, 8, 7, 4, 2, 1

0 1 2 3 4 5 6

a[i] == i



TC : Sorting + Iterating

$O(n \log n) + O(n)$

$\approx O(n \log n)$

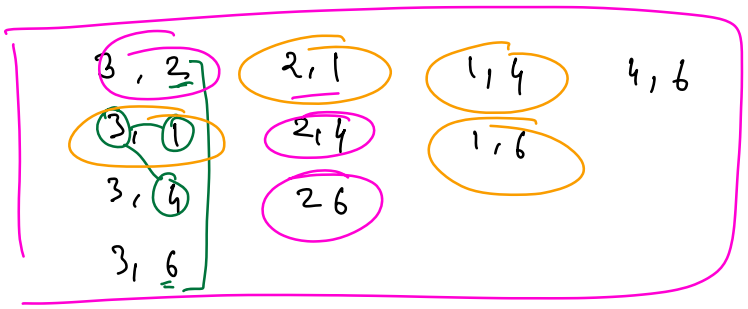
What if --- ? ⇒ Try Out

Q Given an array. Find no. of pairs having the minimum diffn.

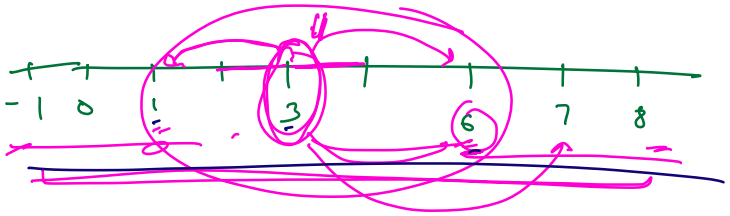
3, 2, 1, 4, 6

{2, 1} {3, 2}, {4, 3} ⇒ 1

⇒ 3



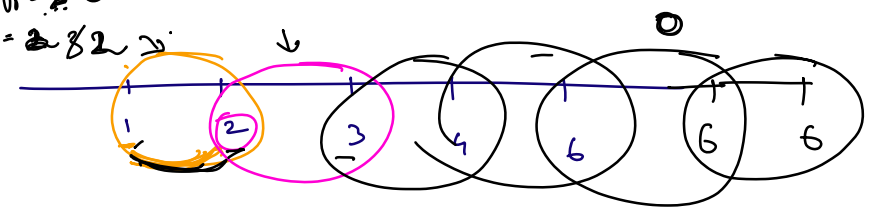
$^nC_2 = O(n^2)$



1, 2, 3, 4, 6

minDiff = ~~0~~

Cnt = ~~2~~ 2




```
Arrays.sort(A);
int minDiff = +∞
int cnt = 0;
for(i=1; i<n; i++){
    if(a[i] - a[i-1] < minDiff){
        minDiff = a[i] - a[i-1];
        cnt = 1;
    }
    else if(a[i] - a[i-1] == minDiff){
        cnt++;
    }
}
```

1000, 2000, 3000

1000

-2 - (-5)

3

Q Amazon MS Adobe

Given an array of 0, 1 & 2.

Sort this array in asc order.

[0, 1, 1, 2, 0, 2, 1, 0]

⇒ library fn] ⇒ O(n log n)

Count Sort

0, 0, 0, 1, 1, 1, 2, 2

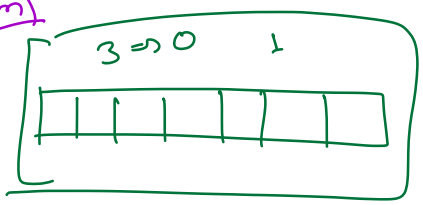
3

```
int countZero = 0;
int countOne = 0;
int countTwo = 0;
for(i=0; i<n; i++){
    if (a[i] == 0)
        countZero++;
    else if (a[i] == 1)
        countOne++;
    else if (a[i] == 2)
        countTwo++;
}
```

O(n)

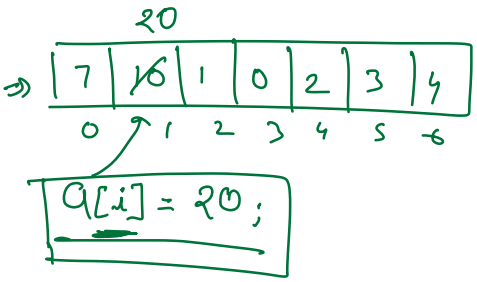
[0, 0, 0, 1, 1, 1, 2, 2]

O(n)



TC : O(N)

SC : O(1)




```
for(i=0; i<n; i++)
    a[i] = 1;
```

1, 1, 1, 1, 1, 1

O(26) ✗

O(1)

O(n)

a x z b c a

a - z

26 { a - 2, b - 1, c - 1, x - 1, ... } => a, a, b, c, x, z

Q Direct in *Amazon Ola

Given an array. (0 - 1)

Find the K^{th} min of the array.

↳ No. at index K in Sorted array → asc

2, 0, 5, 7, 3, 8, 4, 1

K = 3 ⇒ 3

K = 5 ⇒ 5

K = 6 ⇒ 7

2, 1, 2, 1, 2

1, 1, 2, 2, 2

K = 0, 1

1

2

⇒ Sort (O(nlogn))

⇒ a[K]

- Minimize Swaps
- K << n

App 2 Sort only till index K.

How ?

K = 0^{th} ⇒ Smallest / Min Element.

1	3	5	11	7	8	21	32	6	10

TC

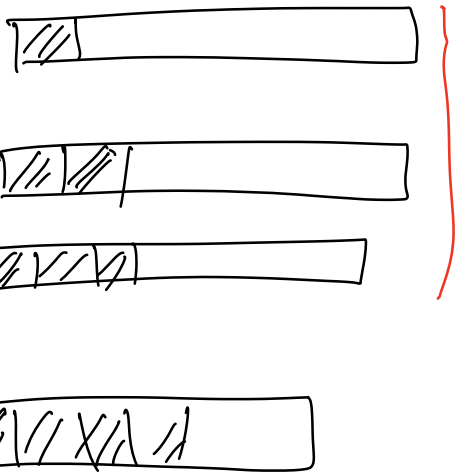
K = 0 → O(N)

1 → O(N)

2 → O(N)

⋮

K → O(N)



⇒ O(KN) ≠ O(n)

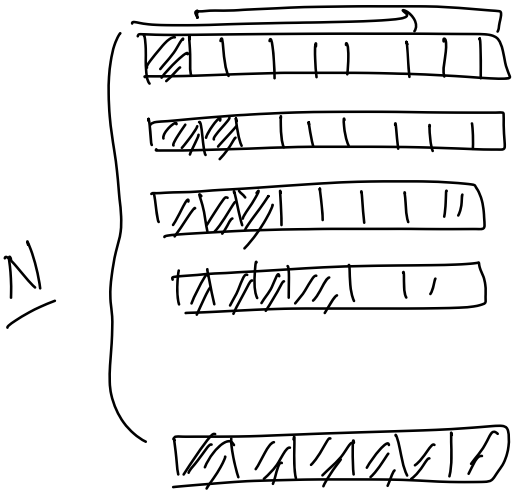
Pseudo Polynomial

K = [0, N-1]

O(N) O(N²) O(log N) O(N log N) O(N³ log N) O(M N)

Repeats n times ⇒ Sorted array

O(n²)



N

SelectionSort

↳ Selecting the min

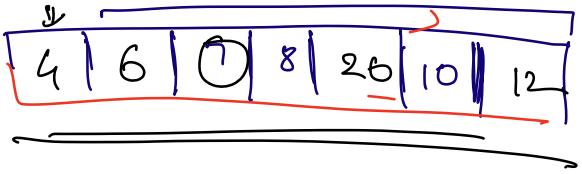
TC : O(N²)

SC : O(1)

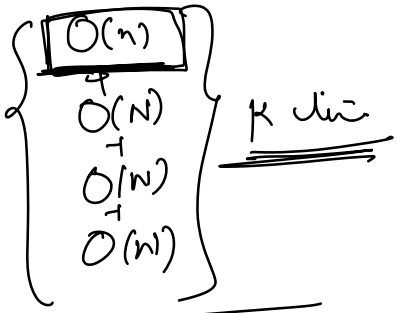
Swaps : N-1

$O(N \log N)$ $O(KN)$

$K < \log N$



$K^{th} = 3$



$O(KN)$

array traversed