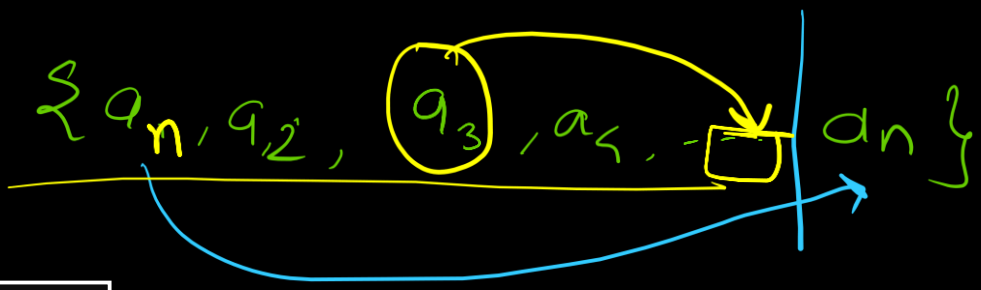
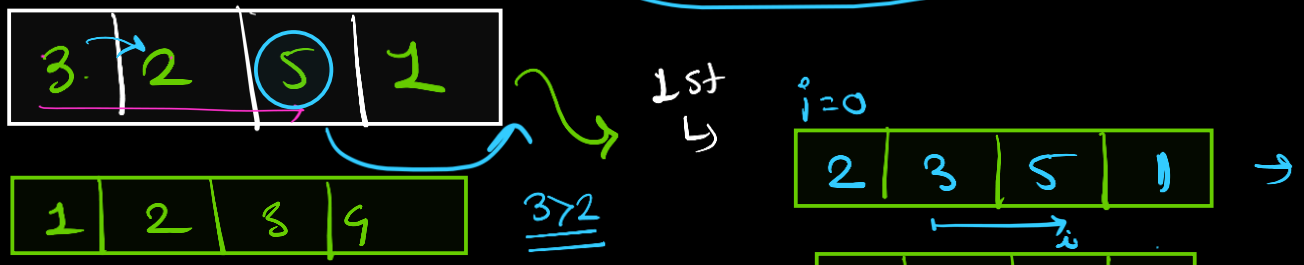
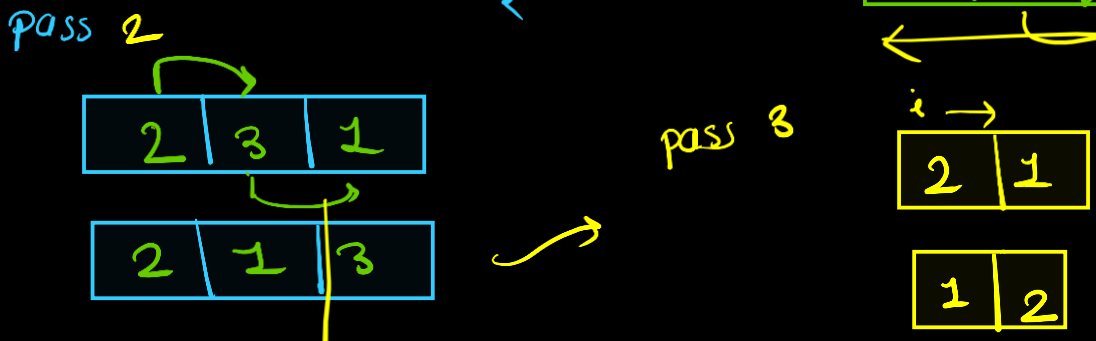
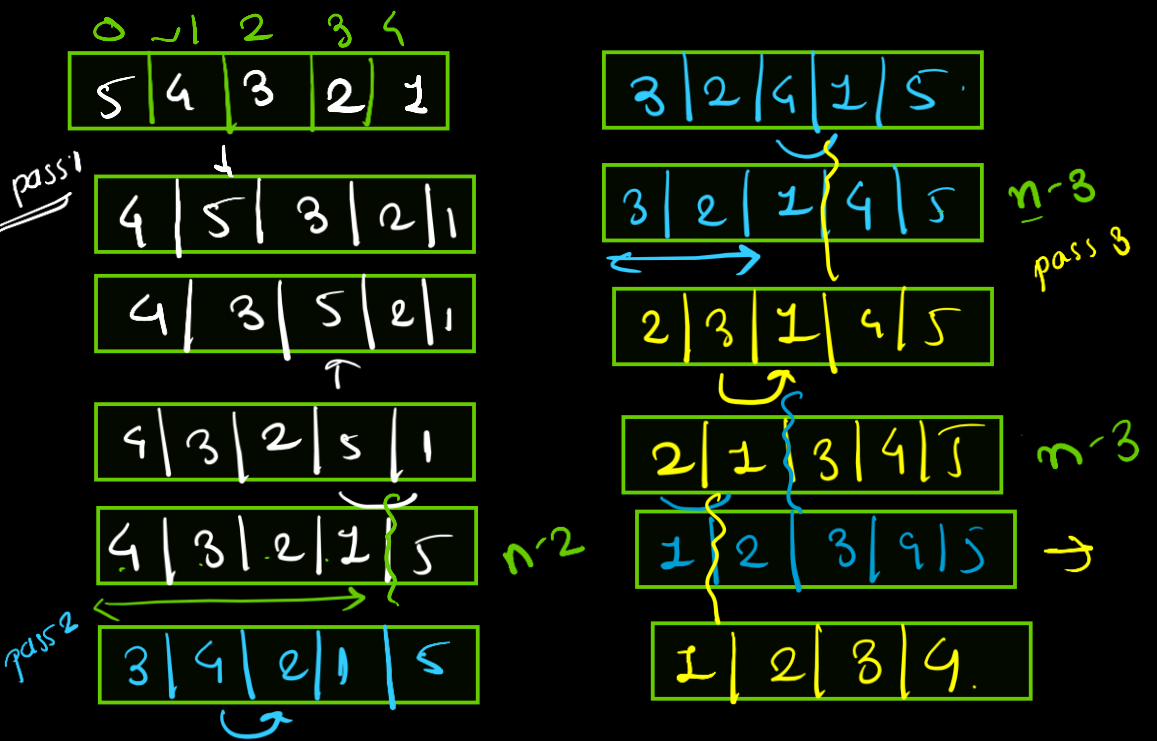
Sorting

→ 1) Bubble sort →









n →

```
for( int i=0; i<n; i++){
    for(int j=0; j<n-i-1; j++){
        if (arr[j] > arr[j+1]) swap(arr[j],
                                    arr[j+1]);
    }
}
```

4st cond

{ a1, a2, a3, a4, a5 }

5 - 0 - 1 = 4

{ a2 a3 a4 a5 | a1 } ⇒ i=0 size - i - 1

{ a2 a4 a5 { a3 a1 } ⇒ sorted

5 - 1 - 1

4 - 1 = 3

1	3	2	4	5

for(i — n) n*n

swap() → swap → sorted

{ 1, 2, 3, 4, 5 }

$2^{31} + 2^{31}$ ⇒

int + int

int int = int

double + float = double

long + int = long

$2^{31} + 1$ =

long ans

int

long =

ans

a b

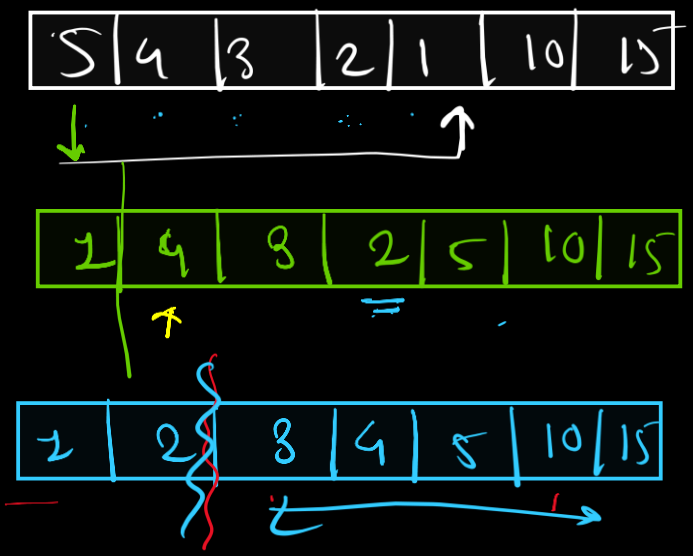
int

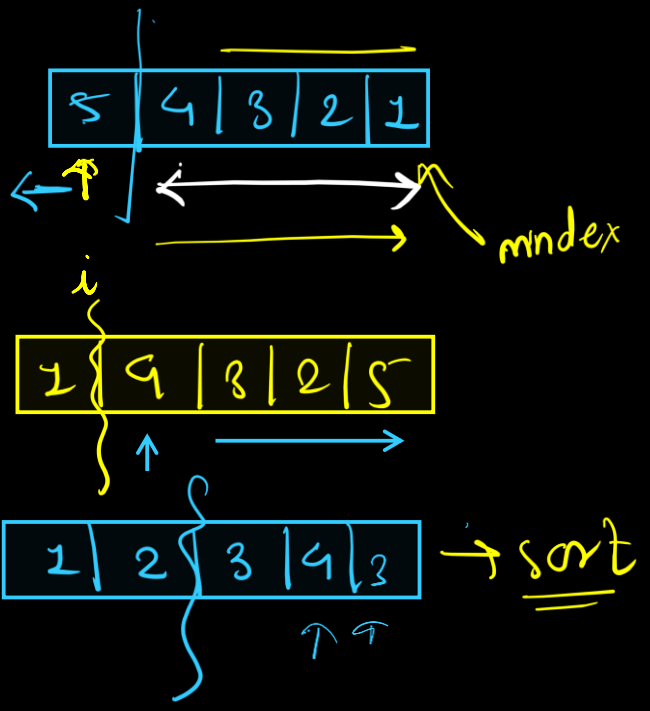
(long)(a + b)

%

Selection Sorting

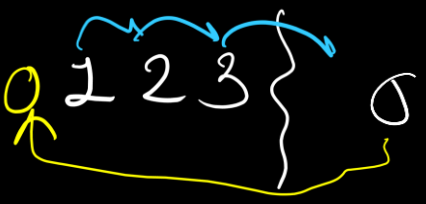
min element first

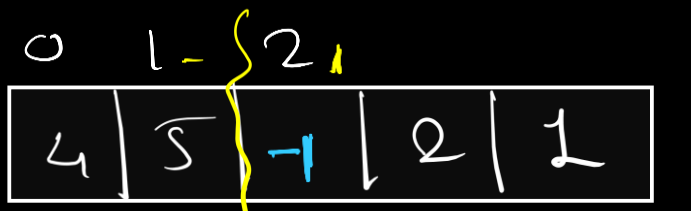




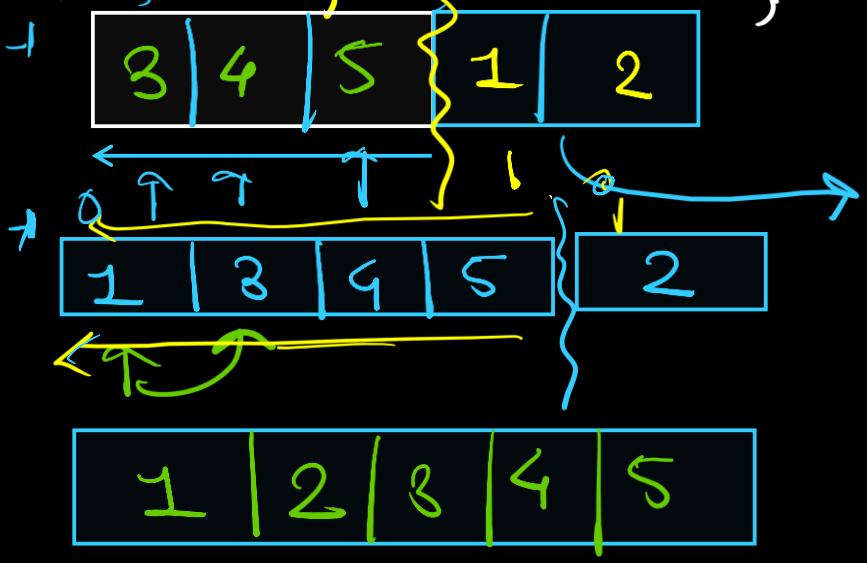

```
for (i=0; i<n; i++){
    minIdxe = i
    for(j=i+1; j<n; j++){
        if(arr[mindexe] > arr[j]){
            minindexe = j
        }
    }
    swap(arr[i], arr[minIndex]);
}
```

Insertion Sort





element = arr[2]



j+1 -1+2=0

1 → 0+1=1

arr[j] > element

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

    int element = arr[i];
    j= i-1;

    while(j>=0){
        if(arr[j] < element) break;
        else arr[j+1] = arr[j]
    }
```

```
    arr[j+1] = element;
}
```