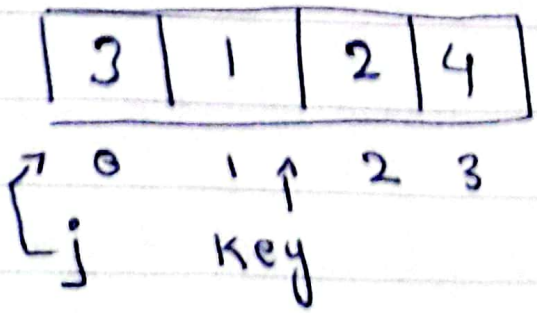
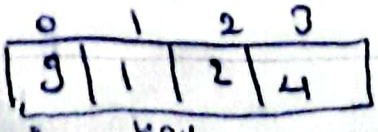
Insertion Sort

Date / /20

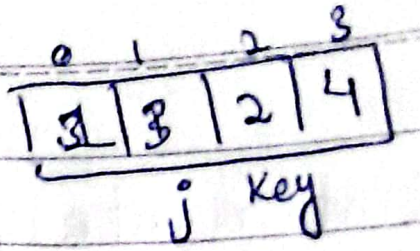
```
Void insertionSort(int *A, int n){
int key, j;
For(int i=1; j<=n-1; i++){
        key = A[i];
        j= i-1;
while(j>=0; && A[j] >key){
        A[j+1] = A[j];
          j--;
    }
     A[j+1]  =key;
    }
}
```



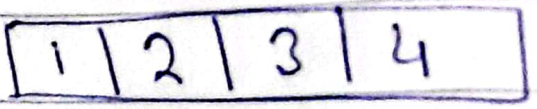
- In 1st iteration — while loop will not be executed.
- In 2nd iteration —



i = 2



j > 0 so while loop will execute:



And further A[j] >! key so the loop will not execute

Selection Sort

```
void selectionSort(int *A, int n){
    int indexofMin, temp;
    for (int i=0; i< n-1; i++){
        indexofMin = i;
        for (int j = i+1; j<n; j++){
            if (A[j] < A[indexofmin]){
                indexofmin = j;
            }
        }
    }
```

temp =
A[i] =
A[ind

3

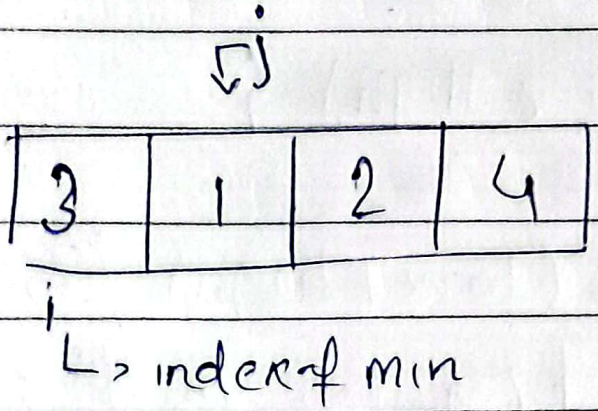
L,

In first

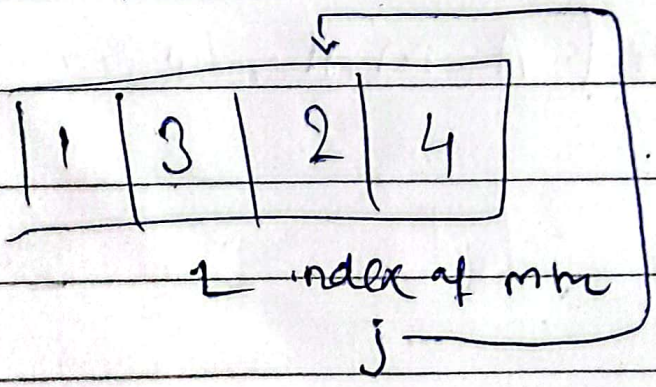
1

and in
con

```
temp = A[i];
A[i] = A[index of min];
A[index of min] = temp;
}
}
```



In first iteration.



1	2	3	4

and in 4th iteration the condition fails.

Merge Sort

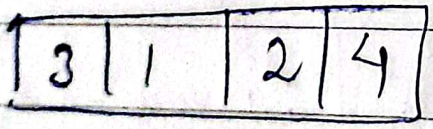
```
void mergeSort (int A[], int low, int
    int mid;
    if(low < High) {
        mid=(low+high)/2;
        mergesort(A, low, mid);
        mergesort(A, mid+1, high);
        merge(A, mid, low, high);
    }
}
```

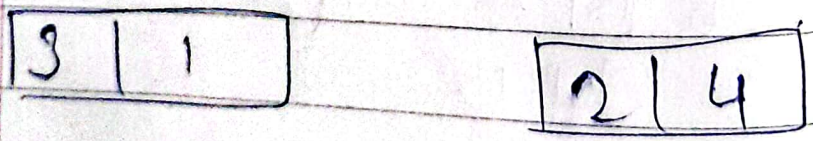
void merge (int A[], int mid, int low, int high){

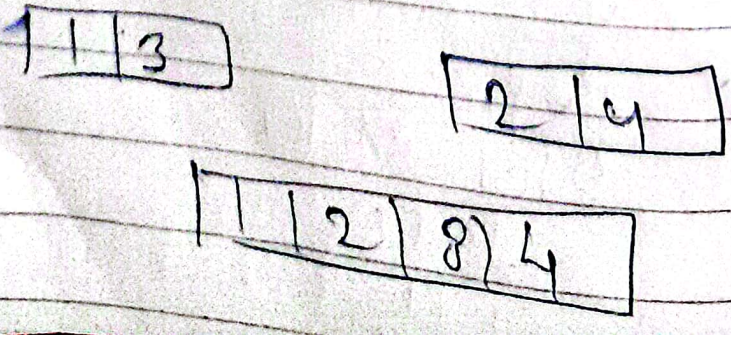
Date: __/__/20

```
int i, j, k, B[200];
i = low;
j = mid + 1;
k = low;

while (i <= mid && j <= high) {
    if (A[i] < A[j]){

        B[k] = A[i];
        i++;
        j++;
    }

else {
    B[k] = A[j];
    j++;
    k++;
    }
while (i <= mid)
    {
    B[k] = A[i];
    k++;
```

```
while (j<=high)
{
    B[k] = A[j];
    k++;
    j++;
}
for (int i=low; i<=high; i++)
{
    A[i] = B[i];
}
}
```







Bubble sort

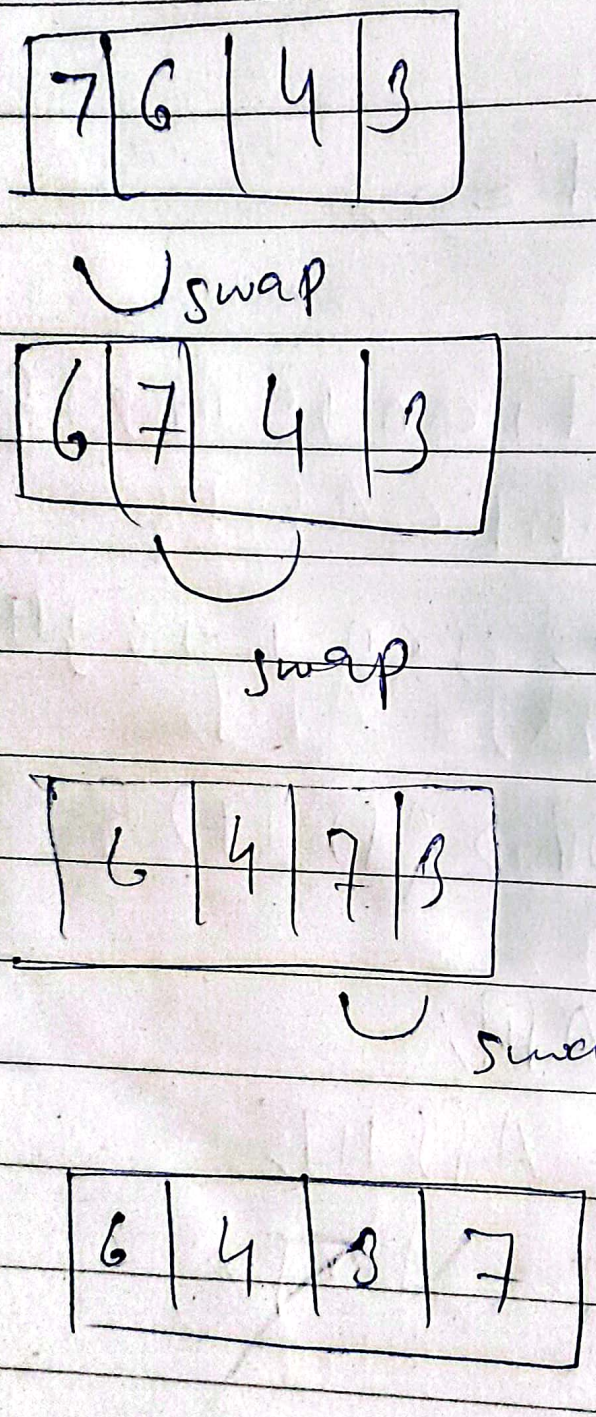
```
void Bubbl
    int temp;
    int isSort
for (int i=0
    for (int j=
        if (A
        temp
        A[j]
        A[j
    }
}
}
```

Bubble sort

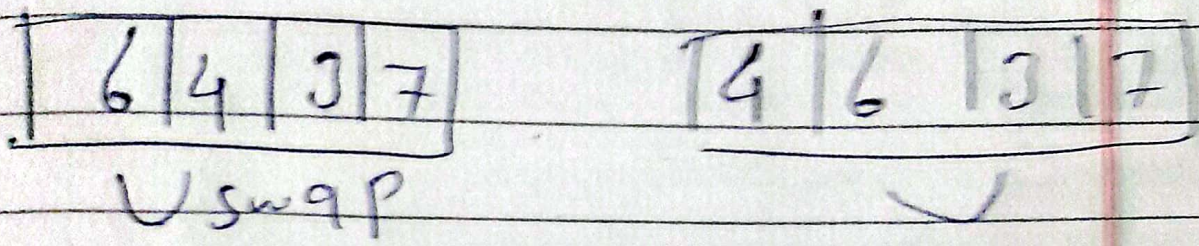
Date: __/__/20

```
void BubbleSort (int *A, int n) {

    int temp;
    int isSorted = 0;

for (int i=0; i<n-1; i++) {

    for (int j=0; j<n-1-i; j++) {

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

        temp = A[j];
        A[j] = A[j+1];
        A[j+1] = temp;
        }
    }
}
}
```

First pass.



Seond pass



4	3	6	7

Third pass