# ASSIGNMENT 4

## 1.CONCURRENT QUICKSORT

```
int partition(int *arr, int l, int r)

{


}


void normal_quickSort(int arr[], int l, int r)

{




}
```

This is normal randomized quicksort algorithm

in which we select a random pivot and then sort the array such that all element less than pivot go to left and others go to right.This is done recursively till the array get sorted.

```
int partition(int *arr, int l, int r)

{


}


void quicksort(int *arr, int l, int r){


}
```

This is Concuurent random Quicksort which finds a random index then we create an array so that all element less than a pivot goes to the left else goes to the right .Then we fork to create two prcess each sort half of the array and then merge them.

```
void *threaded_quicksort(void* a){


}
```

This is a threaded random quicksort it works as the same way as random quicksort except that it create two thread each time to sort half of the given array passed as argument to the parent thread.

Example:

```
6
4 3 1 6 5 1
Running concurrent_mergesort for n = 6
time = 0.000706
1 1 3 4 5 6
Running Threaded_quicksort for n = 6
time = 0.000479
1 1 3 4 5 6
Running normal_quicksort for n = 6
time = 0.000005
1 1 3 4 5 6
normal_mergesort ran:
        [ 135.942995 ] times faster than concurrent_mergesort
        [ 92.103708 ] times faster than threaded_concurrent_mergesort
```

Time taken by threaded_quicksort is less than
concurrent quicksort because in concurrent quicksort we create two process which require th os to make virtual space which is overhead

but in thread we create thread for each process and the share data between each other but this not a case in concurrent process in that we have to create shared memory area so that the process can share data and accessing data will take time so overall concurrent quicksort requires morre time.

The reason why both the above sort are slower as compare to normal quicksort is because here the value of n(vo. of element to be sorted) is small so for small value it is wastage of time and resource for CPU to create thread or forking a process .But as n get larger the become faster as compered to normal quicksort .Threaded quicksort will take less time compared to concurrent quicksort.