Report for question 1

quicksort.c - 2018101069_Assignment4 - Visual Studio Code

```c
#include <inttypes.h>
#include <math.h>

int *shareMem(size_t size)
{
    key_t mem_key = IPC_PRIVATE;
    int shm_id = shmget(mem_key, size, IPC_CREAT | 0666);
    return (int *)shmat(shm_id, NULL, 0);
}

int partition(int arr[], int low, int high)
{
    int pivot = arr[high];
    int i = (low - 1);

    for (int j = low; j <= high - 1; j++)
    {
        if (arr[j] < pivot)
        {
            i++;
            int t = arr[i];
            arr[i] = arr[j];
            arr[j] = t;
        }
    }
    int t = arr[i + 1];
    arr[i + 1] = arr[high];
    arr[high] = t;
    return (i + 1);
}

void quickSort(int arr[], int l, int r)
```

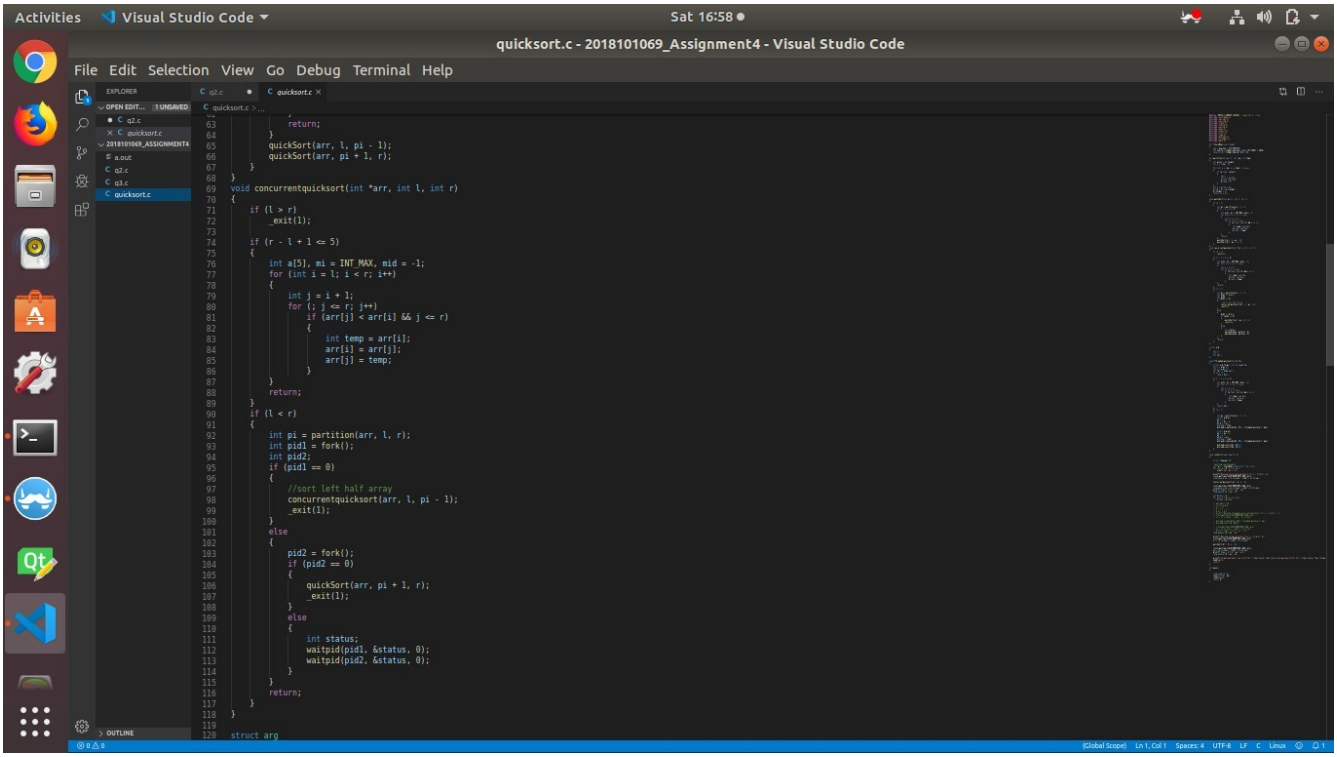
ShareMem fucntion is used to create a shared memory array used in concurrent and threaded quicksort
Partition function is used to get array divide into 2 parts.

quicksort.c - 2018101069_Assignment4 - Visual Studio Code

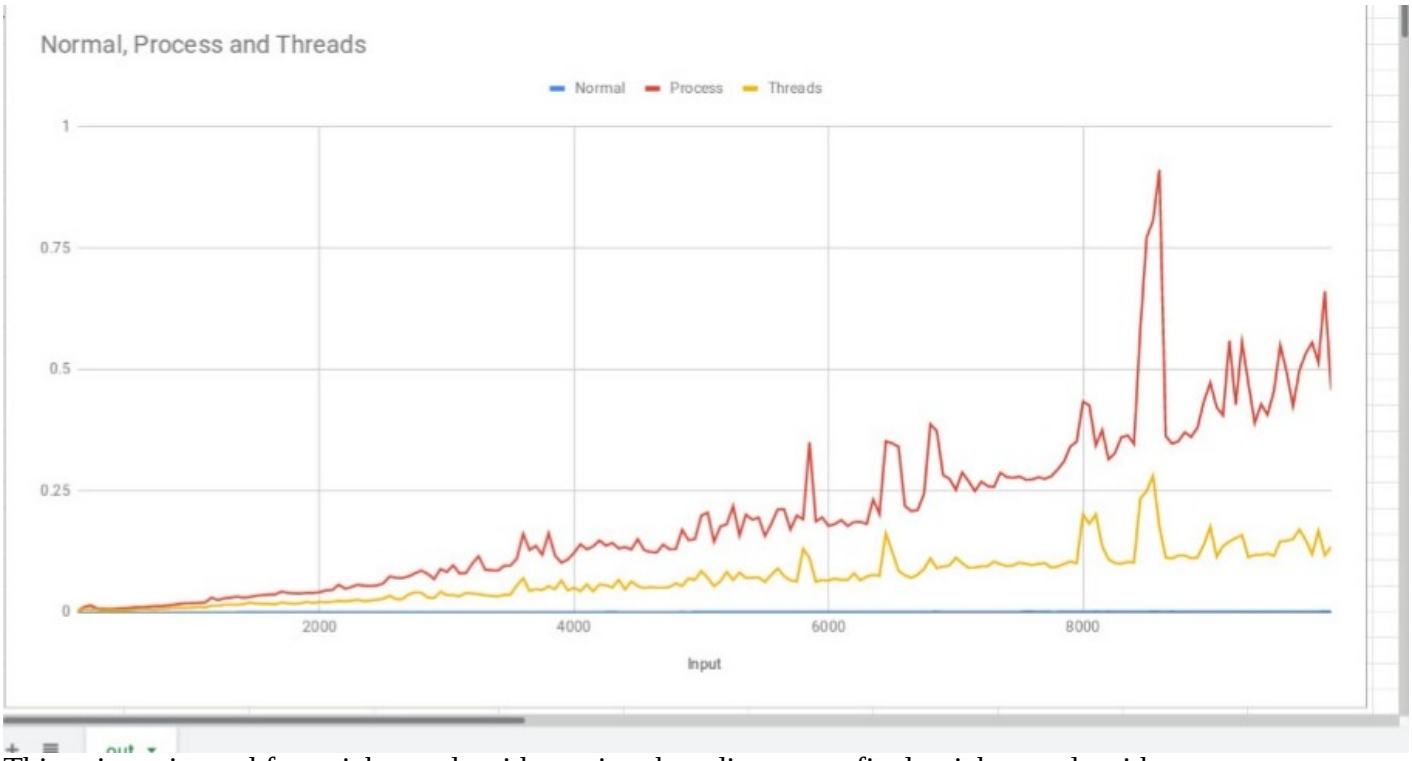
```c
    arr[high] = t;
    return (i + 1);
}

void quickSort(int arr[], int l, int r)
{
    if (l < r)
    {
        int pi = partition(arr, l, r);
        if (r - l + 1 <= 5)
        {
            int a[5], mi = INT_MAX, mid = -1;
            for (int i = l; i < r; i++)
            {
                int j = i + 1;
                for (; j <= r; j++)
                    if (arr[j] < arr[i] && j <= r)
                    {
                        int temp = arr[i];
                        arr[i] = arr[j];
                        arr[j] = temp;
                    }
            }
            return;
        }
        quickSort(arr, l, pi - 1);
        quickSort(arr, pi + 1, r);
    }
}
void concurrentquicksort(int *arr, int l, int r)
{
```

This snippet shows quicksort function which is used to sort using normal quick sort algorithm.

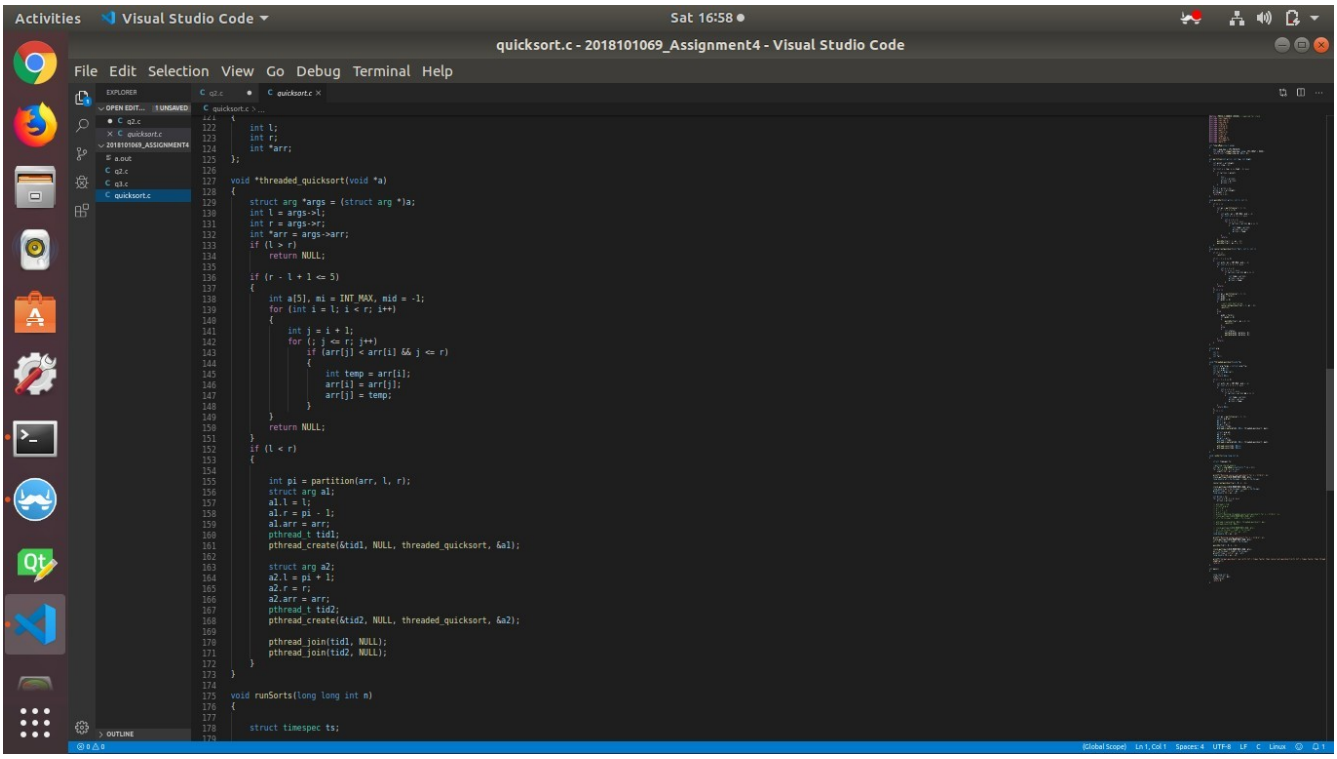


This snippet shows concurrent quicksort function which is used to create parrellel process and run quick sort.



This snippet is used for quicksort algorithm using threading to get final quick sort algorithm

It will create thread to compute sorted array for divided part by giving each thread a subproblem.



This code snippet shows time comparision between normal quicksort and threaded quicksort and concurrent quick sort.