High Performance Computing : Assignment 2

Title:- Parallel Bubble Sort based on existing algorithms using OpenMP

```
#include<iostream>
#include<stdlib.h>
#include<omp.h>
using namespace std;
void bubble(int *, int);
void swap(int &, int &);
void bubble(int *a, int n)
{
   for(  int i = 0;  i < n;  i++ )
    {
          int first = i % 2;

          #pragma omp parallel for shared(a,first)
          for(  int j = first;  j < n-1;  j += 2  )
            {
                   if(  a[ j ]  >  a[ j+1 ]  )
                    {
                            swap(  a[ j ],  a[ j+1 ]  );
                    }
                     }
    }
}
void swap(int &a, int &b)
{
   int test;
   test=a;
   a=b;
   b=test;
}

int main()
{

   int *a,n;
   cout<<"\n enter total no of elements=>";
   cin>>n;
   a=new int[n];
   cout<<"\n enter elements=>";
   for(int i=0;i<n;i++)
   {
          cin>>a[i];
   }

   bubble(a,n);

   cout<<"\n sorted array is=>";
   for(int i=0;i<n;i++)
   {
        cout<<a[i]<<endl;
   }


return 0;
}
```

Output :

```
enter total no of elements=>5

enter elements=>34

21

23

44

12

sorted array is=>12

21

23

34

44
```

Title: Parallel Merge Sort based on existing algorithms using OpenMP

```
#include<iostream>
#include<stdlib.h>
#include<omp.h>
using namespace std;


void mergesort(int a[],int i,int j);
void merge(int a[],int i1,int j1,int i2,int j2);

void mergesort(int a[],int i,int j)
{
        int mid;
        if(i<j)
        {
        mid=(i+j)/2;

        #pragma omp parallel sections
        {
```

```
        #pragma omp section
        {
                mergesort(a,i,mid);
        }

        #pragma omp section
        {
                mergesort(a,mid+1,j);
        }
        }

        merge(a,i,mid,mid+1,j);
        }

}

void merge(int a[],int i1,int j1,int i2,int j2)
{
        int temp[1000];
        int i,j,k;
        i=i1;
        j=i2;
        k=0;

        while(i<=j1 && j<=j2)
        {
        if(a[i]<a[j])
        {
        temp[k++]=a[i++];
        }
        else
        {
        temp[k++]=a[j++];
   }
        }

        while(i<=j1)
        {
        temp[k++]=a[i++];
        }

        while(j<=j2)
        {
        temp[k++]=a[j++];
        }

        for(i=i1,j=0;i<=j2;i++,j++)
        {
        a[i]=temp[j];
        }
```

```
}


int main()
{
        int *a,n,i;
        cout<<"\n enter total no of elements=>";
        cin>>n;
        a= new int[n];

        cout<<"\n enter elements=>";
        for(i=0;i<n;i++)
        {
        cin>>a[i];
        }
        mergesort(a,0,n-1)
        cout<<"\n sorted array is=>";
        for(i=0;i<n;i++)
        {
        cout<<"\n"<<a[i];
        }

        return 0;
}
```

Output:-

```
enter total no of elements=>4
enter elements=>23
34
45
11
sorted array is=>
11
23
34
45
```