Assignment 2

**Bubble sort**

#include<iostream>

#include<stdlib.h>

#include<omp.h>

using namespace std;

void bubblesort(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]);

}

}

// Print array after each iteration

cout << "Array after iteration " << i << ":\n";

for(int k = 0; k < n; k++) {

cout << a[k] << " ";

}

cout << endl;

}

}

void swap(int &a,int &b)

{

int temp=0;

temp=a;

a=b;

b=temp;

}

int main(){

int\* a,n;

cout<<"\n Enter size of Array:";

cin>>n;

a=new int[n];

cout<<"\n Enter elements:\n";

for(int i=0;i<n;i++)

{

cin>>a[i];

}

bubblesort(a,n);

cout<<"\n Sorted array is:\n";

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

cout<<a[i]<<" ";

}

return 0;

}

**Merge sort**

#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;

cout << "\nMerging: ";

for (int x = i1; x <= j1; x++)

{

cout << a[x] << " ";

}

cout << "and ";

for (int x = i2; x <= j2; x++)

{

cout << a[x] << " ";

}

cout << endl;

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];

}

cout << "Result after merging: ";

for (int x = i1; x <= j2; x++)

{

cout << a[x] << " ";

}

cout << endl;

}

int main()

{

int \*a, n, i;

cout << "\nEnter size of Array : ";

cin >> n;

a = new int[n];

cout << "\nEnter elements : \n";

for (i = 0; i < n; i++)

{

cin >> a[i];

}

mergesort(a, 0, n - 1);

cout << "\nSorted array is : ";

for (i = 0; i < n; i++)

{

cout << a[i] << " ";

}

return 0;

}

Run file

g++ -fopenmp program.cpp -o program  
  
./program