Experiment No 2:

MERGE SORT

## *Source Code:*

#include <iostream>

#include <fstream>

#include <limits.h>

#include <cstring>

using namespace std;

class MergeSort{

private:

long int a[11000];

public:

void readInput(int n, int type){

char s[16];

switch (type){

case 0: strcpy(s, "input.txt"); break;

case 1: strcpy(s, "input-asc.txt"); break;

case 2: strcpy(s, "input-desc.txt"); break;

case 3: strcpy(s, "input-equal.txt"); break;

}

ifstream fin(s);

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

fin >> a[i];

fin.close();

}

void writeOutput(int n){

ofstream fout("output.txt");

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

fout << a[n-1-i] << "\t";

fout.close();

}

void mergeSort(int p, int r, int &count){

if (p < r){

int q = (p + r) / 2;

mergeSort(p, q, count);

mergeSort(q+1, r, count);

merge(p, q, r, count);

}

}

void merge(int p, int q, int r, int &count){

int i, j;

int n1 = q - p + 1;

int n2 = r - q;

int L[n1+2], R[n2+2];

for (int i=0; i<n1; i++)

L[i] = a[p+i];

L[n1] = INT\_MAX;

for (int j=0; j<n2; j++)

R[j] = a[q+j+1];

R[n2]=INT\_MAX;

i=0; j=0;

for (int k=p; k<=r; k++){

count++;

if (L[i] <= R[j]){

a[k] = L[i];

i++;

}

else{

a[k] = R[j];

j++;

}

}

}

};

int main(){

MergeSort s;

int ch;

int len[7] = {10, 50, 100, 500, 1000, 5000, 10000};

do {

cout << " 1. Merge Sort, 2. Exit" << endl;

cout << "Enter your option: ";

cin >> ch;

if (ch != 1 && ch==2)

break;

cout << "For Random Input: " << endl;

for (int i=0; i<7; i++){

int count = 0;

int n = len[i];

s.readInput(n, 0);

switch (ch){

case 1: s.mergeSort(0, n-1, count);

break;

}

s.writeOutput(n);

cout << "Input size: " << n << "\t" << "No. of Comparisons: " << count << endl;

}

cout << endl << "For ascending order input: " << endl;

for (int i=0; i<7; i++){

int count = 0;

int n = len[i];

s.readInput(n, 1);

switch (ch){

case 1: s.mergeSort(0, n-1, count);

break;

}

// s.writeOutput(n);

cout << "Input size: " << n << "\t" << "No. of Comparisons: " << count << endl;

}

cout << endl << "For descending order input: " << endl;

for (int i=0; i<7; i++){

int count = 0;

int n = len[i];

s.readInput(n, 2);

switch (ch){

case 1: s.mergeSort(0, n-1, count);

break;

}

// s.writeOutput(n);

cout << "Input size: " << n << "\t" << "No. of Comparisons: " << count << endl;

}

cout << endl << "For all elements are same input: " << endl;

for (int i=0; i<7; i++){

int count = 0;

int n = len[i];

s.readInput(n, 3);

switch (ch){

case 1: s.mergeSort(0, n-1, count);

break;

}

// s.writeOutput(n);

cout << "Input size: " << n << "\t" << "No. of Comparisons: " << count << endl;

}

}

while (ch < 5);

return 0;

}

## *Analysis:*



