

**มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ**

**KING MONGKUT’S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK**

**ASSIGNMENT 4 Quicksort**

**เสนอ**

**อาจารย์ประดิษฐ์ พิทักษ์เสถียรกุล**

**จัดทำโดย**

**นายวรศิษฏ์ ภู่สุวรรณ์**

**ITI-2RB รหัส 6206021421237**

**รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา Data Structure and Algorithm**

**(รหัสรายวิชา 060223119)**

**ภาคการศึกษาที่ 1 ปีการศึกษา 2563**

**สาขา เทคโนโลยีสารสนเทศ ภาควิชา เทคโนโลยีสารสนเทศ**

**คณะเทคโนโลยีและการจัดการอุตสาหกรรม**

**มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ วิทยาเขตปราจีนบุรี**

#include<iostream>

#include<vector>

#include<string>

#include<time.h>

using namespace std;

template<class Comparable>

void insertionsort(vector<Comparable> & a);

template<class Comparable>

void insertionsort(vector<Comparable> & a, int left, int right);

template<class Comparable>

void quicksort(vector<Comparable> & a);

template<class Comparable>

void quicksort(vector<Comparable> &a, int left, int right);

template<class Comparable>

const Comparable &median3(vector<Comparable> &a, int left, int right);

template<class Comparable>

void swap(int &num1, int &num2);

int n = 0;

int cutoff;

//bool flag=true;

void main() {

vector<int> x(1);

string choice;

bool exit = true;

bool create = false;

clock\_t t1, t2, t3, t4;

do {

system("cls");

cout << "\n\t\t\t===============================\n";

cout << "\t\t\t.:: Assignment4 [ Sorting ] ::.\n";

cout << "\t\t\t===============================\n";

cout << "\t\t\t1.) Create Input";

cout << "\n\t\t\t2.) Insertion Sort";

cout << "\n\t\t\t3.) Quick Sort";

cout << "\n\t\t\t4.) Exit";

cout << "\n\t\t\t===============================";

cout << "\n\n\t\t\tEnter Choice : ";

cin >> choice;

switch (choice[0])

{

case '1':

system("cls");

int num;

cout << "\n How many for input : ";

cin >> num;

cout << endl;

x.resize(num);

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

cout << " Input Data [" << i << "] : ";

cin >> x[i];

}

create = !create;

system("pause");

break;

case '2':

system("cls");

if (create)

{

cout << "\n\t\t\tInsertion Sort\n\n";

cout << " Original " << " | ";

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

cout << x[i] << " " ;

cout << "P.Moved\n";

cout << "\n ----------------------------------------------------------" << endl;

t1 = clock();

insertionsort(x);

t2 = clock();

cout << "\n ----------------------------------------------------------" << endl;

cout << "\nRunning Time : " << ((float)t2 - (float)t1) << " second\n" << endl;

system("PAUSE");

}else{

cout << "\n\t\t\t!!!Must Insert before sort!!!\n\n";

system("pause");

}

break;

case '3':

system("cls");

if (create)

{

cout << "\n\t\tEnter Cutoff : ";

cin >> cutoff;

cout << "\n\t\tYour Cutoff is : " << cutoff << endl;

cout << "\n\t\t\tQuick Sort\n\n";

cout << " Original ";

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

cout << x[i] << " ";

cout << "\n ----------------------------------------------------------" << endl;

t3 = clock();

quicksort(x, 0, num - 1);

t4 = clock();

cout << "\n ----------------------------------------------------------" << endl;

cout << "\n Final ";

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

cout << x[i] << " ";

cout << endl;

cout << "\nRunning Time : " << ((float)t4 - (float)t3) << " second\n" << endl;

system("PAUSE");

}else{

cout << "\n\t\t\t!!!Must Insert before sort!!!\n\n";

system("pause");

}

break;

case '4':exit = false;

}

} while (exit);

}

template<class Comparable>

void insertionsort(vector<Comparable> & a) {

int co = 0;

for (int p = 1; p<a.size(); p++) {

int c = 0;

Comparable tmp = a[p];

int j;

for (j = p; j>0 && tmp<a[j - 1]; j--) {

a[j] = a[j - 1];

co++;

c++;

}

a[j] = tmp;

cout << " After p = " << p << " | ";

for (int i = 0; i<a.size(); i++) {

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

}

cout << " " << c << endl;

}

}

template<class Comparable>

void insertionsort(vector<Comparable> & a, int left, int right) {

for (int p = left + 1; p <= right; p++) {

Comparable tmp = a[p];

int j;

for (j = p; j>left&&tmp<a[j - 1]; j--) {

a[j] = a[j - 1];

}

a[j] = tmp;

}

}

template <class Comparable>

void quicksort(vector<Comparable> & a) {

quicksort(a, 0, a.size() - 1);

}

template<class Comparable>

void quicksort(vector<Comparable> &a,int left,int right){

++n;

if(left+cutoff<=right){

Comparable pivot=median3(a,left,right);

int i=left,j=right-1;

for(;;){

while(a[++i]<pivot){}

while(a[--j]>pivot){}

if(i<j)

swap(a[i],a[j]);

else{

break;

}

}

swap(a[i],a[right-1]);

cout<<"\n Round "<<n;

for(int i=0;i<a.size();i++)

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

cout << endl;

cout << "\n Left = " << a[left] << " Right = " << a[right] << " Pivot = " << pivot << endl;

quicksort(a,left,i-1);

quicksort(a,i+1,right);

}

else {

insertionsort(a,left,right);

cout<<"\n Round "<<n;

for (int i = 0; i < a.size(); i++) {

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

}

cout<<" No Pivot "<<endl;

}

}

template<class Comparable>

const Comparable &median3(vector<Comparable> &a,int left,int right){

int center=(left+right)/2;

if(a[center]<a[left])

swap(a[left],a[center]);

if(a[right]<a[left])

swap(a[left],a[right]);

if(a[right]<a[center])

swap(a[right],a[center]);

swap(a[center],a[right-1]);

return a[right-1];

}

template<class Comparable>

void swap(int &num1,int &num2){

int tmp=num1;

num1=num2;

num2=tmp;

}