Dustin O'Brien

COSC 320-001 Lab 01 2/7/2023

Lab Report: I have completed I finished on my own but did look at previous assignments to

remind myself syntax of the makefile, main function, and including a header file to main file. It took Approximately 2 hours to code but most of it was due to realizing I misunderstood some piece of assignment like the need for a header instead of including selection sort function in the main function. In reference to the written work that took me 3-4 hours to complete most of which came in figuring out how to figure out the time complexity of the program.

Lab 1.0

Source Code:

#include "deSelSort.h"

int main ()

{

int size = 8;

int array[] = {13, 5, 2, 25, 47, 17, 8, 21};

deSelsort (array,size);

return 0;

}

#ifndef deSelSort\_H

#define deSelSort\_H

#include <iostream>

using namespace std;

void print\_Array(int array[] , int size)

{

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

{

cout<<array[i]<<" ";

}

cout<<endl;

}

void deSelsort (int array[] , int size)

{

int Minimum\_Array\_Location = 0;

int Maximum\_Array\_Location = 0;

int Temporary= 0;

for (int i=0; i<size/2;i++)

{

print\_Array(array,size);

Minimum\_Array\_Location = i;

Maximum\_Array\_Location = i;

for (int j=i; j<size-i;j++)

{

if(array[Maximum\_Array\_Location]<array[j])

{

Maximum\_Array\_Location=j;

}

if(array[Minimum\_Array\_Location]>array[j])

{

Minimum\_Array\_Location=j;

}

}

Temporary = array[i];

array[i]=array[Minimum\_Array\_Location];

array[Minimum\_Array\_Location] = Temporary;

Temporary = array[size-i-1];

array[size-i-1]=array[Maximum\_Array\_Location];

array[Maximum\_Array\_Location] = Temporary;

}

print\_Array(array,size);

}

#endif

Output 1:

13 5 2 25 47 17 8 21

2 5 13 25 21 17 8 47

2 5 13 8 21 17 25 47

2 5 8 13 17 21 25 47

2 5 8 13 17 21 25 47

Output 2:

13 5 2 25 47 17 8 21

2 5 13 25 21 17 8 47

2 5 13 8 21 17 25 47

2 5 8 13 17 21 25 47

2 5 8 13 17 21 25 47

Output 3:

13 5 2 25 47 17 8 21

2 5 13 25 21 17 8 47

2 5 13 8 21 17 25 47

2 5 8 13 17 21 25 47

2 5 8 13 17 21 25 47

size = 8;

int array[] = {13, 5, 2, 25, 47, 17, 8, 21};

deSelsort (array,size);

return 0;

}

Questions:

1. My algorithm will first allocate memory and set 3 variables than inside the nested for loop that runs from size - 2 \* number of times the outer loop has run the algorithm

does 2 checks one if Maximum array value is greater than the value that its currently in and then checks if the same value is the minimum value so far if one of those returns

true it'll update the respective value then in the outer loop it'll swap min with first value of non sorted array and max with last value of non sorted array



2.

Text, letter

Description automatically generated

As seen in work above which should represent the worst case the algorithm has a time complexity of O(n^2) which is the same as the normal selection algorithm, so it isn’t more efficient.