Kevin Vo

Dr. Lehr

Course: CSC-17C

Assignment: Operational Analyses of Problems 1, 2, 3

Date: 5/2/18

**Problem #1 Linear Search – Show O( ) operation analysis**

\*\*\*\* Problem 1 - Linear Search \*\*\*\*

Let N = 1000000

Value = -1 was not found!

Runtime = 0.009999 seconds

Number of Compare Operations = 3e+06

Number of Equal (assigning values) Operations = 3

Number of Incrementing Operations = 1e+06

Total Number of Operations = 4e+06

------------------------------------------------------------------------------------------------------------------------------------------

Let N = 2000000

Value = -1 was not found!

Runtime = 0.0225748 seconds

Number of Compare Operations = 9e+06

Number of Equal (assigning values) Operations = 6

Number of Incrementing Operations = 3e+06

Total Number of Operations = 1.2e+07

------------------------------------------------------------------------------------------------------------------------------------------

Let N = 4000000

Value = -1 was not found!

Runtime = 0.0506697 seconds

Number of Compare Operations = 2.24076e+07

Number of Equal (assigning values) Operations = 9

Number of Incrementing Operations = 7e+06

Total Number of Operations = 2.94076e+07

------------------------------------------------------------------------------------------------------------------------------------------

Let N = 8000000

Value = -1 was not found!

Runtime = 0.0774537 seconds

Number of Compare Operations = 3.35544e+07

Number of Equal (assigning values) Operations = 12

Number of Incrementing Operations = 1.5e+07

Total Number of Operations = 4.85544e+07

**Problem #1 Binary Search – Show O( ) operation analysis**

\*\*\*\* Problem 1 - Binary Search \*\*\*\*

Let N = 1000000

--- Timer Started ---

--- Timer Ended ---

Runtime for Binary Search = 8.8e-06 seconds

That number does not exist in the array.

Number of Compare Operations = 78

Number of Math Operations (+, -, \*, /, =) = 100

Number of Incrementing Operations = 0

Total Number of Operations = 178

------------------------------------------------------------------------------------------------------------------------------------------

Let N = 2000000

--- Timer Started ---

--- Timer Ended ---

Runtime for Binary Search = 2e-06 seconds

That number does not exist in the array.

Number of Compare Operations = 82

Number of Math Operations (+, -, \*, /, =) = 105

Number of Incrementing Operations = 0

Total Number of Operations = 187

------------------------------------------------------------------------------------------------------------------------------------------

Let N = 4000000

--- Timer Started ---

--- Timer Ended ---

Runtime for Binary Search = 2.7e-06 seconds

That number does not exist in the array.

Number of Compare Operations = 86

Number of Math Operations (+, -, \*, /, =) = 110

Number of Incrementing Operations = 0

Total Number of Operations = 196

------------------------------------------------------------------------------------------------------------------------------------------

Let N = 8000000

--- Timer Started ---

--- Timer Ended ---

Runtime for Binary Search = 2.7e-06 seconds

That number does not exist in the array.

Number of Compare Operations = 90

Number of Math Operations (+, -, \*, /, =) = 115

Number of Incrementing Operations = 0

Total Number of Operations = 205

**Problem #2 Bubble Sort – Show O( ) operation analysis**

\*\*\*\* Problem 2 - Bubble Sort \*\*\*\*

Let N = 10000

Assigning random integers to be sorted...

Sorting values...

--- Timer has Started ---

--- Ending has Ended ---

Array has been sorted with Bubble Sort.

Runtime = 1

Number of Arithmetic Operations = 1.34218e+08

Number of Compare Operations = 1.67772e+07

Number of Increment Operations = 1.67772e+07

Total Numbers of Operation = 1.67772e+08

------------------------------------------------------------------------------------------------------------------------------------------

Let N = 20000

Assigning random integers to be sorted...

Sorting values...

--- Timer has Started ---

--- Ending has Ended ---

Array has been sorted with Bubble Sort.

Runtime = 1

Number of Arithmetic Operations = 1.34218e+08

Number of Compare Operations = 1.67772e+07

Number of Increment Operations = 1.67772e+07

Total Numbers of Operation = 1.67772e+08

------------------------------------------------------------------------------------------------------------------------------------------

Let N = 40000

Assigning random integers to be sorted...

Sorting values...

--- Timer has Started ---

--- Ending has Ended ---

Array has been sorted with Bubble Sort.

Runtime = 10

Number of Arithmetic Operations = 1.34218e+08

Number of Compare Operations = 1.67772e+07

Number of Increment Operations = 1.67772e+07

Total Numbers of Operation = 1.67772e+08

------------------------------------------------------------------------------------------------------------------------------------------

Let N = 80000

Assigning random integers to be sorted...

Sorting values...

--- Timer has Started ---

--- Ending has Ended ---

Array has been sorted with Bubble Sort.

Runtime = 24

Number of Arithmetic Operations = 1.34218e+08

Number of Compare Operations = 1.67772e+07

Number of Increment Operations = 1.67772e+07

Total Numbers of Operation = 1.67772e+08

------------------------------------------------------------------------------------------------------------------------------------------

**Problem #2 Selection Sort – Show O( ) operation analysis**

\*\*\*\* Problem 2 - Selection Sort \*\*\*\*

Let N = 10000

Assigning Values to Array to be sorted by Selection Sort...

Sorting Array with Selection Sort...

--- Timer has started. ---

--- Timer has ended. ---

Selection Sort completed.

Runtime = 0 seconds

Number of Compare Operations = 1.68137e+07

Number of Arithmetic (+, - , \*, /, =) Operations = 3.35544e+07

Number of Incrementing Operations = 1.67772e+07

Total Number of Operations = 6.71454e+07

----------------------------------------------------------------

Let N = 20000

Assigning Values to Array to be sorted by Selection Sort...

Sorting Array with Selection Sort...

--- Timer has started. ---

--- Timer has ended. ---

Selection Sort completed.

Runtime = 2 seconds

Number of Compare Operations = 1.68555e+07

Number of Arithmetic (+, - , \*, /, =) Operations = 3.35544e+07

Number of Incrementing Operations = 1.67772e+07

Total Number of Operations = 6.71872e+07

----------------------------------------------------------------

Let N = 40000

Assigning Values to Array to be sorted by Selection Sort...

Sorting Array with Selection Sort...

--- Timer has started. ---

--- Timer has ended. ---

Selection Sort completed.

Runtime = 7 seconds

Number of Compare Operations = 1.69364e+07

Number of Arithmetic (+, - , \*, /, =) Operations = 3.35544e+07

Number of Incrementing Operations = 1.67772e+07

Total Number of Operations = 6.7268e+07

----------------------------------------------------------------

Let N = 80000

Assigning Values to Array to be sorted by Selection Sort...

Sorting Array with Selection Sort...

--- Timer has started. ---

--- Timer has ended. ---

Selection Sort completed.

Runtime = 28 seconds

Number of Compare Operations = 1.70968e+07

Number of Arithmetic (+, - , \*, /, =) Operations = 3.35544e+07

Number of Incrementing Operations = 1.67772e+07

Total Number of Operations = 6.74284e+07

**Problem #3 Simple Vector (Inefficient) – Show O( ) operation analysis**

\*\*\*\* Problem 3 - Simple Vector (Inefficient)

Let N = 1000

Runtime = 2

Number of Compare Operations = 500500

Number of Increment Operations = 502500

Number of Equal Operations = 505500

Total Operations = 1508500

---------------------------------------------------------

Let N = 2000

Runtime = 7

Number of Compare Operations = 2001000

Number of Increment Operations = 2005000

Number of Equal Operations = 2011000

Total Operations = 601700

---------------------------------------------------------

Let N = 4000

Runtime = 27

Number of Compare Operations = 8002000

Number of Increment Operations = 8010000

Number of Equal Operations = 8022000

Total Operations = 24034000

---------------------------------------------------------

Let N = 8000

Runtime = 109

Number of Compare Operations = 32004000

Number of Increment Operations = 32020000

Number of Equal Operations = 32044000

Total Operations = 96068000

**Problem #3 Simple Vector (Efficient) – Show O( ) operation analysis**

\*\*\*\* Problem 3 - Simple Vector (Efficient)

Let N = 1000

Timer has Started ---

Timer has Ended ---

Runtime for Simple Vector Efficient = 0.0106518 seconds

Number of Compare Operations = 2023

Number of Increment Operations = 2023

Number of Equal Operations = 2083

Number of Arithmetic (+, /, \*, -) Operations = 10

Total Operations = 6139

Let N = 2000

Timer has Started ---

Timer has Ended ---

Runtime for Simple Vector Efficient = 0.0164313 seconds

Number of Compare Operations = 4047

Number of Increment Operations = 4047

Number of Equal Operations = 4113

Number of Arithmetic (+, /, \*, -) Operations = 11

Total Operations = 12218

Let N = 4000

Timer has Started ---

Timer has Ended ---

Runtime for Simple Vector Efficient = 0.0365118 seconds

Number of Compare Operations = 8095

Number of Increment Operations = 8095

Number of Equal Operations = 8167

Number of Arithmetic (+, /, \*, -) Operations = 12

Total Operations = 24369

Let N = 8000

Timer has Started ---

Timer has Ended ---

Runtime for Simple Vector Efficient = 0.067362 seconds

Number of Compare Operations = 16191

Number of Increment Operations = 16191

Number of Equal Operations = 16269

Number of Arithmetic (+, /, \*, -) Operations = 13

Total Operations = 48664

**Problem #3 Simple Vector (Linked List) – Show O( ) operation analysis**

\*\*\*\* Problem 3 - Simple Vector (Linked List)

Let N = 1000

Timer has Started ---

Timer has Ended ---

Runtime for Simple Vector Linked List = 0.0109261 seconds

Number of Compare Operations = 500500

Number of Increment Operations = 0

Number of Equal Operations = 1506500

Total Operations = 2007000

--------------------------------------------------------

Let N = 2000

Timer has Started ---

Timer has Ended ---

Runtime for Simple Vector Linked List = 0.0323279 seconds

Number of Compare Operations = 2001000

Number of Increment Operations = 0

Number of Equal Operations = 6013000

Total Operations = 8014000

--------------------------------------------------------

Let N = 4000

Timer has Started ---

Timer has Ended ---

Runtime for Simple Vector Linked List = 0.128223 seconds

Number of Compare Operations = 8002000

Number of Increment Operations = 0

Number of Equal Operations = 24026000

Total Operations = 32028000

--------------------------------------------------------

Let N = 8000

Timer has Started ---

Timer has Ended ---

Runtime for Simple Vector Linked List = 0.539581 seconds

Number of Compare Operations = 32004000

Number of Increment Operations = 0

Number of Equal Operations = 96052000

Total Operations = 128056000

--------------------------------------------------------