**Johnathan Soto**

**CS 315 Data Structures Spring 2018**

Lab 3 solution

**Problems:**

1)

1. How many times is the instruction **B[j] += B[k]** executed?

Answer = (99\*98) / 2 = ***4,851*** times B[j] += B[k] is executed

1. What is the number of times that the instruction B[j] = 0 is executed?

Answer = ***99 times***, B[j] = 0 is executed

2) How many multiplications are performed by the above code segment ?

Answer: (3 for loops multiplication) 24,000 + (2 for loop multiplication) 1,200 = ***25,200***

***multiplications are preformed***

3)

The number of times each operation is performed:

* Assignments: 1999
* Comparisons: 1997
* Additions: 1497
* Multiplications: 499
* Modulo: 998
* Increment: 999
* Subtractions: 1497
* Total: 9486

4)

The number of key comparisons: (best case) 3n/2

(worst case) 3n/2

The number of key assignments: (best case) 2

(worst case) 5n/2

5)

1. Determine the ***exact*** number of ***key comparisons*** performed by the above program in the ***worst-case***.

Answer: ((n - 1) \* n) / 2 = ((n^2) – n)/ 2 OR O(N^2)

1. Do the same thing for the ***best-case***.

Answer: (n – 1) \* 1 = n – 1 OR O(N)

1. Assuming that the keys used are the integers 1 to *n*, what input is the best-case input?

Answer: The best-case would be going from 1 to n increasing in value till integer n, starting at integer 1. (eg: 1,2,3,4)

1. Assuming that the keys used are the integers 1 to *n*, what input is the worst-case input?

Answer: The worst-case is going from n to 1 decreasing from n to 1, starting at integer n. (e.g: 4,3,2,1)

6) Recall the recursive algorithm for computing x that you wrote for lab # 1:

1. Compute the number of multiplications performed by the above function for *n* = 1000, 2000, 3000, …, 10000 and display the result in a table.

|  |  |
| --- | --- |
| n | multiplications |
| 1000 | 14 |
| 2000 | 15 |
| 3000 | 17 |
| 4000 | 16 |
| 5000 | 16 |
| 6000 | 18 |
| 7000 | 18 |
| 8000 | 17 |
| 9000 | 17 |
| 10000 | 17 |

1. *M*(*n*) is shown below:

*M*(*n*) = 0 if *n* = 0 or 1

  M(n/2) + 1  if *n* is even and *n* > 0

M((n-1) / 2) + 1 if *n* is odd and *n* > 1

1. n worst-case average-case

100 4,950 2,524.5

200 19,900 10,049.5

300 44,850 22,574.5

400 79,800 40,099.5

500 124,750 62,624.5

600 179,700 90,149.5

700 244,650 122,674.5

800 319,600 160,199.5

900 404,550 202,724.5

1000 499,500 250,249.5

(Here include the result from wolfram-alpha that reports the values of *a*, *b* and *c*.)

Average A(n) Formula: 0.25(n^2) + (7.59433x10^(-13))(n) – 9.0555x10^(-9)

a: 0.25 b: 7.59433x10^(-13) c: – 9.0555x10^(-9)

n worst-case average-case (rounded)

100 4,950 2,500

200 19,900 10,000

300 44,850 22,500

400 79,800 40,000

500 124,750 62,500

600 179,700 90,000

700 244,650 122,500

800 319,600 160,000

900 404,550 202,500

1000 499,500 250,000