

Fundamentals of Algorithms

Lab 1

David Hill, Jr.

Due: 8/28/2017

1.) Copy and paste the lines of code from your implementation of bubblesort that demonstrates the initialization step of the loop invariant? (10 points)

```
37 void bubbleSort(vector<string> &strArr)
38 {
39     int size = strArr.size();           // Initilization step of loop invariant.
40 }
```

2.) Copy and paste the lines of code from your implementation of bubblesort that demonstrates the maintenance step of the loop invariant? (10 points)

```
42     for (int x = 0; x < size; x++)      // Maintenance steps
43     {
44         for (int y = 0; y < size - x - 1; y++)
45         {
46             int L1 = int(tolower(strArr[y][0]));
47             int L2 = int(tolower(strArr[y+1][0]));
48
49             if (L1 > L2)
50             {
51                 swap(strArr[y], strArr[y+1]);
52             }
53         }
54     }
55
56                                     // End of maintenance steps
57 }
```

3.) Copy and paste the lines of code from your implementation of bubble sort that demonstrates the termination step of the loop invariant? (10 points)

```
104
105     cout << " Unsorted: " ;
106
107     for(int idx = 0; idx < names; idx++)    // Termination steps
108     {
109         cout << " " << B[idx] << " ";
110     }
111
112     bubbleSort(B);
113
114     cout << endl;
115
116     cout << " Sorted: ";
117
118     for(int idx = 0; idx < names; idx++)    // End of termination steps
119     {
120         cout << " " << B[idx] << " ";
121     }
122
123 }
```

4.) What is the purpose of identifying the three steps of the loop invariant in your code? What does it prove? (10 points)

The purpose of identifying the loop invariant steps in your code is to validate the algorithm code to ensure it meets the requirements of a loop invariant. Moreover, this proves the correctness of the algorithm being analyzed.

5.) Provide the GitHub link for your Yourfirstname_lab1.cpp file. (50 points)

https://github.com/dhillii/Fundamentals-of-Algorithms/tree/master/David_Lab1

6.) What do you think was the most difficult aspect of this code lab? (10 points)

The hardest part of this lab was modifying the original bubble sort function to work with a vector of strings. The challenge in this was finding the correct logic to swap strings in the vector based on the “if” condition. Also, figuring out how to compare strings to see which one was alphabetically first was a challenge. This was solved by comparing ASCII values.