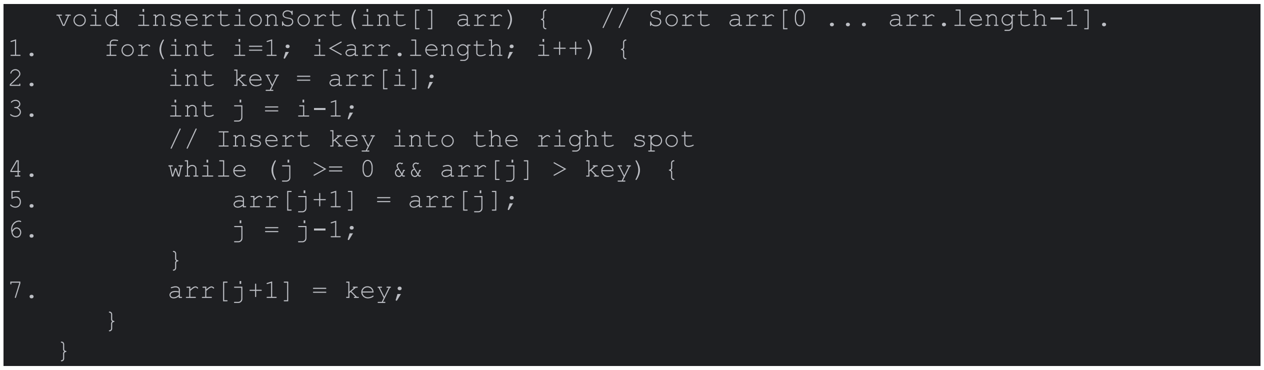
Assignment 2

Name: Sai Naga Viswa Chaitanya, Basava
Email: sainagaviswachaitanya.basava@utdallas.edu
NetID: SXB220302

Assuming all basic operations take a constant time of c = 1.

Insertion sort:



```
   void insertionSort(int[] arr) {   // Sort arr[0 ... arr.length-1].
1.    for(int i=1; i<arr.length; i++) {
2.        int key = arr[i];
3.        int j = i-1;
          // Insert key into the right spot
4.        while (j >= 0 && arr[j] > key) {
5.            arr[j+1] = arr[j];
6.            j = j-1;
          }
7.        arr[j+1] = key;
      }
   }
```

Analysis:

line 1 – $1 + (\mathrm{n} + 1) + n = 2n + 2$

line 2 – n

line 3 – n

line 7 – n

The execution of the inner loop (lines 4,5,6) is going to be dependent on the distribution of elements in the array. In worst case, it will take i iterations for each outer loop iteration (array is arranged in reverse order) and in best case it can take 0 iterations each (array already sorted or element already in the right position).

Best case runtime analysis:

Line 4,5,6 – 0

Total = $5n + 2$

Runtime complexity = $O(n)$

Worst case runtime analysis:

Line 4,5,6 – $\sum_{i=1}^{n-1} 3i = 1.5n(n-1)$

Total = $1.5n^2 + 3.5n + 2$

Runtime complexity = $O(n^2)$