L5 - Insertion Sort

Tuesday, August 18, 2020 9:49 AM

Conceptual Framework:

1. Divide the list into two parts, sorted part & unsorted part
2. Insert one element from the unsorted part into the sorted part
   a. Find the correct position in the sorted part to insert the element
   b. Make room for the new element
   c. Place the new element in the correct position
3. Repeat step two to until the entire list is sorted

5 2 4 6 3 1

6 5 4 3 2 1

```
k  j
1  2  3  4  5  6      sorted list  1 .. j - 1
5  2  4  6  1  3      key = 2

   k  j
1  2  3  4  5  6      sorted list  1 .. j - 1
2  5  4  6  1  3      key = 4

      k  j
1  2  3  4  5  6      sorted list  1 .. j - 1
2  4  5  6  1  3      key = 6

         k  j
1  2  3  4  5  6      sorted list  1 .. j - 1
2  4  5  6  1  3      key = 1

      k     j
1  2  3  4  5  6      key = 1
2  4  5  6  6  3

   k        j
1  2  3  4  5  6      key = 1
2  4  5  5  6  3
```

```
InsertionSort(Arr)
1. for j=2 to Arr.length
2.     let key = Arr[j]
3.     let k = j - 1
4.     while k > 0 and Arr[k] > key
5.         Arr[k+1] = Arr[k]
6.         k = k - 1
7.     Arr[k+1] = key
```

```
k       j
1  2  3  4  5  6      key = 1
2  4  4  5  6  3
```

```
k          j
1  2  3  4  5  6      key = 1
2  2  4  5  6  3
```

```
k          j
1  2  3  4  5  6      key = 1
1  2  4  5  6  3      Arr[k+1]=key
```

```
      k  j
1  2  3  4  5  6      sorted list  1 .. j - 1
1  2  4  5  6  3      key = 3
```

```
   k        j
1  2  3  4  5  6      key = 3
1  2  3  4  5  6      Arr[k+1] = key
```

```
                j
1  2  3  4  5  6      sorted list  1 .. j - 1
1  2  3  4  5  6
```

InsertionSort(Arr)	cost	freq
1. For j=2 to Arr.length	c1	n
2. let key = Arr[j]	c2	n-1
3. let k = j - 1	c3	n-1
4. while k > 0 and Arr[k] > key	c4	(n+2)(n-1)/2 (n-1)
5. Arr[k+1] = Arr[k]	c5	n(n-1)/2 (0)
6. k = k - 1	c6	n(n-1)/2 (0)
7. Arr[k+1] = key	c7	n-1

Worst case Analysis:
Worst Case occurs when the input list is already sorted in the reverse order of the desired order.

6 5 4 3 2 1 (List already sorted in reverse order)
5 6 4 3 2 1 (1+1)
4 5 6 3 2 1 (2+1)
3 4 5 6 2 1 (3+1)
2 3 4 5 6 1 (4+1)
1 2 3 4 5 6 (5+1)

Taking Sum
2+3+4+5+…+(n-1)+n
(n+2)(n-1)/2

For lines 5 & 6, the series is 1+2+3+4+5+…+n-1=n(n-1)/2

T(n) = c1(n) + (c2+c3+c7)(n-1)/2 + (c5+c6)(n(n-1)/2) + c4(n+2)(n-1)/2
T(n) = c10(n^2) +c11n + c12
$T_w(n) = O(n^2)$

Best case:
Best Case occurs when the list is already sorted in the desired/required order.

1 2 3 4 5 6 (List already sorted in required order.)
1 2 3 4 5 6 (1)
1 2 3 4 5 6 (1)
1 2 3 4 5 6 (1)
1 2 3 4 5 6 (1)
1 2 3 4 5 6 (1)

T(n) = c1(n) + (c2+c3+c4+c7)(n-1)
T(n) = c11n + c12
$T_B(n) = \Omega(n)$

$T(n) = \Theta(n^2)$ X

$T(n) = \Theta(n)$ X