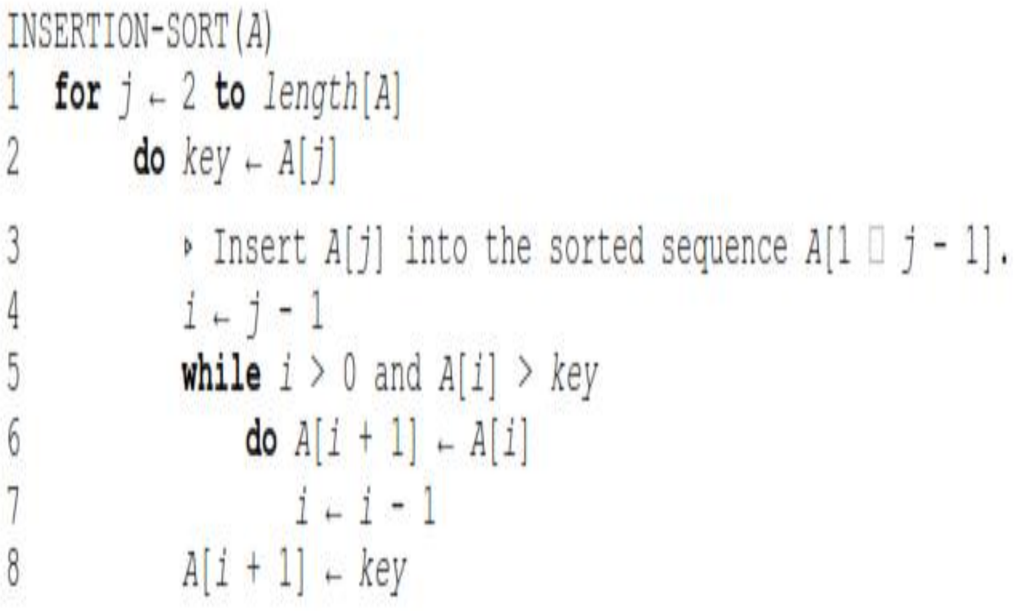
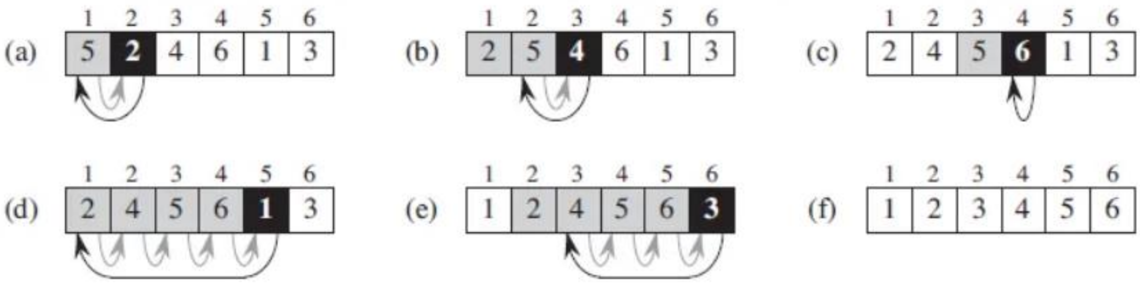
Unit-1: Analysis of Sorting Algorithm

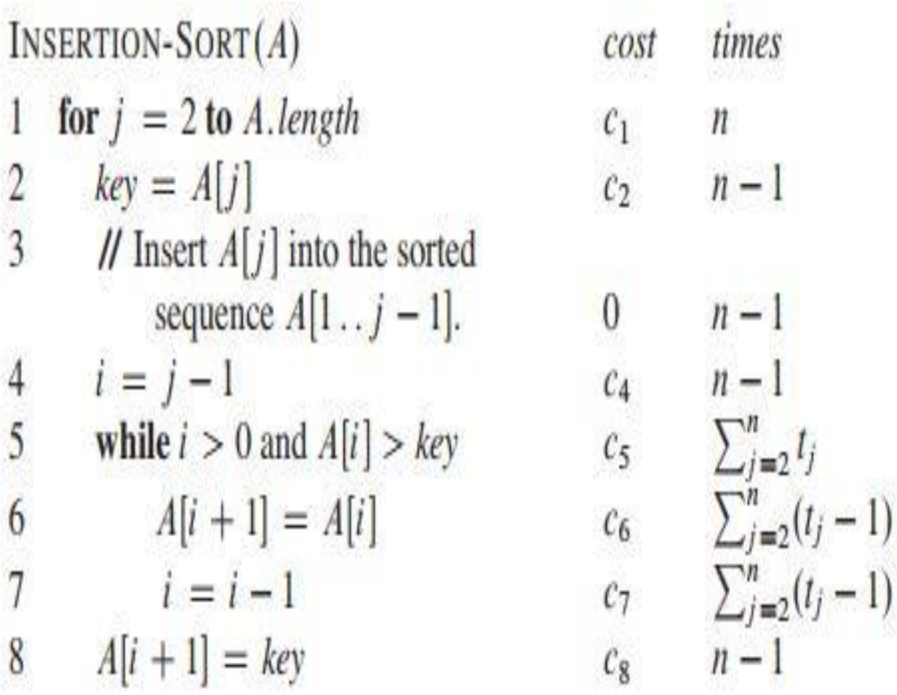
Insertion Sorting Algorithm:



```
INSERTION-SORT(A)
for j ← 2 to length[A]
    do key ← A[j]
       ▹ Insert A[j] into the sorted sequence A[1 □ j - 1].
       i ← j - 1
       while i > 0 and A[i] > key
           do A[i + 1] ← A[i]
              i ← i - 1
       A[i + 1] ← key
```

A= [5, 2, 4, 6, 1, 3]





```
INSERTION-SORT(A)                                  cost    times
1  for $j = 2$ to $A.length$                       $c_1$   $n$
2      $key = A[j]$                                $c_2$   $n-1$
3      // Insert $A[j]$ into the sorted
           sequence $A[1..j-1]$.                   0       $n-1$
4      $i = j - 1$                                 $c_4$   $n-1$
5      while $i > 0$ and $A[i] > key$              $c_5$   $\sum_{j=2}^{n} t_j$
6          $A[i+1] = A[i]$                         $c_6$   $\sum_{j=2}^{n} (t_j - 1)$
7          $i = i - 1$                             $c_7$   $\sum_{j=2}^{n} (t_j - 1)$
8      $A[i+1] = key$                              $c_8$   $n-1$
```

A= [5, 2, 4, 6, 1, 3]

Best case running time (when the sequence of numbers are already sorted):

$$t_j = 1$$

Then the lines in the while loop will not be executed. Therefore the running time is,

$$T(n) = C_1 n + C_2(n-1) + C_3(n-1) + C_4(n-1) + C_7(n - 1)$$

Equals to:

$$T(n) = (C_1 + C_2 + C_3 + C_4 + C_7)\, n - (C_2 + C_3 + C_4 + C_7)$$

Thus a linear function of **n.**

Worst case scenario of the insertion sort (When the sequence of numbers are in the reverse sorted order). Then,

$$t_j = j$$

since we have to go backward till the beginning every time (have to loop back to the first number of the array **A** for every **j**). Thus, while loop condition will be checked for

$$\sum_{j=2}^{n} j = \frac{n(n+1)}{2} - 1$$

times (equals to 2 + 3 + ... + n). And

$$\sum_{j=2}^{n} (j-1) = \frac{n(n-1)}{2}$$

Then the total running time in the worst case:

$$T(n) = C_1 n + C_2 (n - 1) + + C_3(n - 1) + C_4 (\frac{n(n+1)}{2} - 1) + C_5 (\frac{n(n-1)}{2}) + C_6 (\frac{n(n-1)}{2}) + C_7(n - 1)$$

That equals to:

$$T(n) = (\frac{C4+C5+C6}{2}) n^2 + (C_1 + C_2 + C_3 + \frac{C4-C5-C6}{2} + C_7) n - (C_2 + C_3 + C_4 + C_7)$$

Simply an equation of **an**^{2} + **bn** + **c** (quadratic function). Therefore if we consider the rate of growth or the order of growth of the running time, we have to only consider the term **an**^{2} since the other terms become insignificant for large values of **n**. The constant **a** also becomes insignificant when the **n** is large. So, we say that insertion sort has a worst case running time of θ(n^2) {theta of n-squared}.