Time Complexity Analysis of Insertion sort

The first loop will always execute n-1 times .

Worst Case : if the array is reversed ordered the inner while loop will execute once when I=1 , twice when I=2 ……., n-1 times when I=n-1.

```
int insertionsort(int arr[],int n)          Cost          time
{

   int i,j,key;
   for(i=1;i<n;i++)
   {
      key=arr[i];                           c1            n-1
      j=i-1;
      while(j>=0 && arr[j]>key)
      {
         arr[j+1]=arr[j];                   c2            (1+2+3+…+n-1)
         j=j-1;
      }
      arr[j+1]=key;                         c3            n-1


   }
```

}

$T(n) = (c1+c3)n-1+c3(1+2+3.....+n-1)$

$= (c1+c3)n-1 + c3 . n(n-1)/2$

$=an^2+b^2+c$

$= O(n^2)$

Average case : $O(n^2)$

Best case: when the array is allredy sorted. The inner loop wont execute.

$T(n) = (c1+c3)n-1$

$= (c1+c3)n-1$

$=an+b$

$= O(n)$