**Recursive Insertion Sort**

Insertion sort can be expressed as a recursive procedure as follows: In order to sort A[0..n-1], we recursively sort A[0..n-2] and then insert A[n-1] into the sorted array A[0..n-2].

1. Write a recursive procedure to implement this version of the Insertion sort.

|  |
| --- |
| void InsertionSort(int A[], int N) {  if(N <= 1) //Base case  return;    InsertionSort(A, N-1); //Sort first N-1 elements    int item = A[N-1]; //Insert last element at correct position  int j = N-2;  while(j>=0 && A[j]>item) {  A[j+1] = A[j];  j--;  } //end while  A[j+1] = item;  } //end InsertionSort |

1. Write the recurrence that expresses the running time of your algorithm.

|  |  |  |
| --- | --- | --- |
| T(n) = |  | 1 ,n <= 1  T(n-1)+n ,n>1 |
|  |

1. Express the running time in **big-O notation** by solving the recurrence using repeated expansions.

T(n) = T(n-1) + n T(n)=T(n-1)+n

T(n-1)=T(n-1-1)+n-1

T(n-2)=T(n-2-1)+n-2

= [ T(n-2) + (n-1) ] + n

= [ T(n-3) + (n-2) ] + (n-1) + n

.

.

.

T(n-k) +

n – k = 1 => k = n – 1

T(n) = T( n - (n-1) ) +

= T(1) +

= 1 + n + (n-1) + (n-2) + … + 2 + 1

= = + = O()