Nicolás Ramírez Calderón
257084

Average Time of Insertion Sort for Permutation with Repetition

Así se calculaba el tiempo de ejecución.

	Cost	Times
procedure insertion_sort(A[1...n])		
for i ← to n do	c_1	n
x ← A[i]	c_2	n - 1
☐ insert A[i] into A[1,..,I-1]	c_3	0
j ← i-1	c_4	n - 1
while x<A[j] and j>0 do	c_5	$\sum_{i=2,..,n} t_i$
A[j+1] ← A[j]	c_6	$\sum_{i=2,..,n} (t_i - 1)$
j ← j-1	c_7	$\sum_{i=2,..,n} (t_i - 1)$
end_while		
A[j+1] ← x	c_8	n - 1
end_for		
end		

Best case time complexity $t_b(n)$

In this case $t_i = 1$ then

$$t_b(n) = c_1 n + (c_2 + c_4 + c_8)(n-1) + c_5 \sum_{i=2,..,n} 1 + (c_6 + c_7) \sum_{i=2,..,n} 0$$

$$= (c_1 + c_2 + c_4 + c_8 + c_5)\, n - (c_2 + c_4 + c_8 + c_5)$$

$$= an + b$$

$$= \Theta(n)$$

Worst case time complexity t_w (n)

In this case $t_i = i$ then

$$t_w(n) = c_1 n + (c_2 + c_4 + c_8)(n-1) + c_5 \sum_{i=2,..,n} i + (c_6 + c_7) \sum_{i=2,..,n} (i-1)$$

$$= an^2 + bn + c$$

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

Average case time complexity t_a (n)

We know that the average case time complexity is like:

$$t_a(n) = an^2 + bn + c$$

Se puede aproximar t_a con los valores obtenidos en uno de los Applets construidos, dando los siguientes resultados.

$$t_a(n) = 0{,}75\,n^2 + 3{,}5\,n - 3$$

$$t_a(n) = \frac{3}{4}n^2 + \frac{7}{2}n - 3$$