	custo	quantidade
1. função OrdenaPorInserção(vetor[]:int)		
2. para j = 2 até tamanho(vetor)		
3. chave = vetor[j]		
4. i = j-1		
5. enquanto i > 0 e vetor[i] > chave		
6. vetor[i+1] = vetor [i]		
7. i = i-1		
8. vetor [i+ 1] = chave		

Resposta:

	custo	quantidade
1. função OrdenaPorInserção(vetor[]:int)		
2. para j = 2 até tamanho(vetor)	op1	n
3. chave = vetor[j]	op2	n-1
4. i = j-1	op3	n-1
5. enquanto i > 0 e vetor[i] > chave	op4	$\sum_{j=2}^{n} tj$
6. vetor[i+1] = vetor [i]	op5	$\sum_{j=2}^{n} (tj-1)$
7. i = i-1	op6	$\sum_{j=2}^{n} (tj-1)$
8. vetor [i+ 1] = chave	op7	n-1

Obs.: Considere o termo tj representando o número de vezes da execução do teste do loop interno. Este valor irá variar de acordo com o melhor e o pior caso.
Visão inicial dos custos de processamento:

T(n)= n op1 + (n-1) op2 + (n-1) op3 + $\sum_{j=2}^{n} tj$ op4 + $\sum_{j=2}^{n} (tj - 1)$ op5 + $\sum_{j=2}^{n} (tj - 1)$ op6 + (n-1) op7

O Melhor caso ocorre quando o vetor está ordenado. Neste caso, para cada valor de j=2, 3, ...,n o teste da chave com o valor i do vetor (vetor[i]>chave) será verdadeiro quando i tem o valor inicial igual j-1. Deste modo conclui-se que $t_j=1$ para j=2, 3, ..., n. Assim os somatórios serão resolvidos como abaixo, para os valores de N e (n-1), sendo que a estimativa do tempo de execução resulta linear em n.

$\sum_{j=2}^{n} tj = (n-1)$ $\quad\quad$ $\sum_{j=2}^{n} (tj - 1) = 0$

T(n)= n op1 + (n-1) op2 + (n-1) op3 + (n-1) op4 + (n-1) op7

T(n)= n op1 + n op2 – op2 + n op3 -op3 + n op4 - op4 + n op7 - op7

T(n)= n (op1 + op2 + op3 + op4 + op7) – (op2 + op3 + op4 + op7)

O pior caso ocorre quando o vetor estiver ordenado de forma inversa. Neste caso será realizada a comparação do elemento vetor[i] e da chave para todos os elementos do vetor em análise (ou seja, o vetor[1...j-1]), portanto $t_j=j$ para j=2, 3, .., n. Assim temos:

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

Neste caso,

T(n)= n op1 + (n-1) op2 + (n-1) op3 + $\sum_{j=2}^{n} tj$ op4 + $\sum_{j=2}^{n} (tj - 1)$ op5 + $\sum_{j=2}^{n} (tj - 1)$ op6 + (n-1) op7

T(n)= n op1 + n op2 - op2 + n op3 - op3 + $\left[\frac{n(n+1)}{2} - 1\right]$ op4 + $\left[\frac{n(n+1)}{2}\text{-n}\right]$op5+ $\left[\frac{n(n+1)}{2}\text{-n}\right]$op6+n op7 - op7

T(n)= n op1 + n op2 - op2 + n op3 - op3 + $\left[\frac{n(n+1)}{2} - 1\right]$ op4 + $\left[\frac{n(n+1)}{2}\text{-n}\right]$op5+$\left[\frac{n(n+1)}{2}\text{-n}\right]$op6+ n op7 - op7

T(n) = $\left[\frac{op4}{2} + \frac{op5}{2} + \frac{op6}{2}\right]$ n^2 + $\left[op1 + op2 + op3 + \frac{op4}{2} + \frac{op5}{2} + \frac{op6}{2} + op7\right]$ n – (op2+op3+op4+op7)

A estimativa do tempo de execução é quadrática T(n) = n^2