

Ejercicio 2

```
void ejercicio2 (vector < int > & v){
    int j, i=1, n=v.size();
    if (n>1) do{
        int x = v[i];
        for (j=i; j > 0 and v[j-1] > x; j--)
            v[j]=v[j-1];
        v[j]=x;
        i++;
    } while (i<n);
}
```

Primero de todo definimos la complejidad del problema por el tamaño del vector "v" que viene representado por "n" una vez visto eso, buscamos el mejor y peor caso:

Encontramos un mejor caso, en el bucle de la línea 5, si lo que hay dentro en vector es mayor a la lo que hay en la posición marcada por el bucle anterior no se hace el bucle, entonces si siempre se cumple eso nunca iterara el bucle, solo se haría el bucle exterior (línea 3-9) que al incrementarse de 1 en 1 el contador su orden es n. Por lo tanto, $\Omega(n)$.

En el peor caso como el programa tendría que iterar los dos bucles, como hemos dicho el bucle exterior se incrementa de 1 en 1, y el bucle interior decrementamos del 1 en 1 hasta llegar a 0, sin embargo "j" (que controla el bucle interior) depende de "i" (la del bucle exterior), por lo tanto tendríamos la siguiente relación:

iteración	i	j
1	1	1
2	2	2
3	3	3
.	.	.
n	n-1	n-1

La combinada de las dos seria estaría definida por:

$$\sum_{i=1}^{n-1} i \in O(n^2)$$

$O(n^2)$ seria su orden.