Insertion sort algorithm

1. Assume elements zero (0) through *p*-1 are sorted (in ascending order).
2. Insert the element in position *p* into its proper place in the preceding (sorted) elements. Now *p*+1 elements are sorted.
3. Perform steps 1 and 2 above starting with element 0 being the entire sorted array from 0 to *p*-1. Element 1 is in position *p*. Increment *p* after each iteration of steps 1 and 2 until all the elements in the array have been sorted.

Original	**34**	**8**	**64**	**51**	**32**	**21**	**Positions Moved**
After p =1	8	34	64	51	32	21	1
After p = 2	8	34	64	51	32	21	0
After p =3	8	34	51	64	32	21	1
After p = 4	8	32	34	51	64	21	3
After p = 5	8	21	32	34	51	64	4

Table 1. Insertion sort after each pass

Worst case analysis:

1. When *p*=1, the element in position *p* could be shifted to position 0, requiring one shift.
2. When *p*=2, the element in position *p* could be shifted to position 0, requiring two shifts.
3. When *p*=3, the element in position *p* could be shifted to position 0, requiring three shifts.
4. When *p*=*N*-1, the element in position *p* could be shifted to position 0, requiring *p* shifts.

Sum all shifts.

$$\sum_{p=1}^{N-1} p = \frac{N(N-1)}{2} = \frac{1}{2}N^2 - \frac{1}{2}N = O(N^2)$$