`Insert(`$A[1..n]$`: array;` ℓ `index in` A`)`
`@1` $x \leftarrow A[\ell]$
`@2` $i \leftarrow \ell - 1$
`@3` `while` $i > 0$ `and` $A[i] > x$ `do`
`@4` $\quad A[i+1] \leftarrow A[i]$
`@5` $\quad i \leftarrow i - 1$
`@6` `done`
`@7` $A[i+1] \leftarrow x$

Дефиниция 0.1. *Нека $A[1..n]$ е масив от числа, а $\ell \in \{1, 2, \ldots, n\}$. Дефинираме множествата от положителни и отрицателни инверсии на ℓ в A като:*

$$\begin{aligned} \sigma^-(A, \ell) &= \{i < \ell \,|\, A[i] > A[\ell]\} \\ \sigma^+(A, \ell) &= \{i > \ell \,|\, A[i] \leq A[\ell]\}. \end{aligned}$$

Лема 0.1. *Нека $A[1..n]$ е масив от числа, $\ell \leq n$ и $A[1..\ell-1]$ е сортиран във възходящ ред. Тогава $Insert(A[1..n], \ell)$ завършва за време $\Theta\left(|\sigma^-(A, \ell)|\right)$ и резултатният масив A' е пермутация на A, за която:*

1. *$A'[\ell+1..n] = A[\ell+1..n]$.*
2. *$A'[1..\ell]$ е сортиран във възходящ ред.*

Доказателство. Операциите на редове @1, @2 и @7 са атомарни. Нещо повече, ясно е, че $x = A[\ell]$ и то не се променя в хода на изпълнение на процедурата. Да разгледаме while-цикъла и да забележим, че при всяко изпълнение на тялото му (редове) @4 и @5 i намалява с 1. Нека A_i е масивът A непосредствено преди изпълнението на ред @3 с параметър i. Ще покажем, че:

$I(i) : A_i[\ell{+}1..n] = A[\ell{+}1..n]$ и $A_i[1..i] = A[1..i]$ и $A_i[i{+}2..\ell] = A[i{+}1..\ell{-}1]$ и $(i < \ell{-}1 \Rightarrow (A_i[i{+}1] = A[i{+}1] \,\&\, x < A[i{+}1]))$.

Правим индукция по i за $i \geq 0$:

- $i = \ell - 1$. $A_{\ell-1}[1..\ell-1] = A[1..\ell-1]$, защото редовете @1 и @2 не променят масива. Останалата част на $I(\ell-1)$ е тривиално вярно (защото $i + 1 > \ell - 1$ и съответно не е вярно, че $i < \ell - 1$).
- Нека е налице $I(i)$ и да допуснем, че $i > 0$. Тогава, ако се изпълни тялото на цикъла още веднъж, тоест A_{i+1} е дефинирано, то това означава, че $x < A_i[i]$, което от $I(i)$ означава, че $x < A[i] = A[(i-1)+1]$. С това вторият конюнкт от импликацията е изпълнен. Да забележим също, че на ред @4 се променя $A_i[i+1]$ и съответно получаваме $A_{i-1}[i+1] = A_i[i] \stackrel{I(i)}{=} A[i]$. Останалите елементи на A_i и A_{i-1} са равни, тоест $A_i[k] = A_{i-1}[k]$ за $k \neq i$. Следователно:

  $$A_{i-1}[1..i-1] = A_i[1..i-1] \stackrel{I(i)}{=} A[1..i-1] \text{ и } A_{i-1}[i{+}1..\ell] = A_{i-1}[i{+}1] \circ A_i[i{+}2..\ell] = A[i] \circ A_i[i{+}2..\ell] \stackrel{I(i)}{=} A[i..\ell-1].$$

  Накрая остана да проверим, че $A_{i-1}[i] = A_i[i] = A[i]$.

С това показахме, че ако процедурата завърши, то наистина при стойност i $A_i[1..i] = A[1..i]$, $A_i[i{+}2..\ell] = A[i{+}1..\ell{-}1]$ и $A[\ell] = x < A[i{+}1] = A_i[i{+}2]$. Тъй като цикълът завършва, то $i = 0$, тоест $A'[1..n] = x \circ A_i[2..n]$ или $x \geq A_i[i] = A[i]$ и $A'[1..n] = A[1..i] \circ x \circ A[i+1..\ell-1] \circ A[\ell+1..n]$. И в двата случая $A'[1..\ell]$ е сортирана пермутация на $A[1..\ell]$ и $A[\ell+1..n] = A'[\ell+1..n]$.

Накрая, остана да забележим, че всяка итерация съответства на индекс $i < \ell$, за който $A[i] = A_i[i] > x = A[\ell]$, тоест всяка итерация съответства на точно една инверсия от $\sigma^-(A, \ell)$. Следователно, while-цикълът ще изпълни $2|\sigma^-(A, \ell)| + 3$ операции и поради това ще завърши и неговата времева сложност ще бъде $\Theta(|\sigma^-(A, \ell)|)$. □

`InsertSort(`$A[1..n]$ `: array)`
`@1` `for` $\ell = 1$ `to` n `do`
`@2` $\quad$`Insert(`A, ℓ`)`
`@3` `done`

Теорема 0.1. *При всеки вход $A[1..n]$, процедурата $InsertSort(A)$ връща сортирана пермутация A' на A. Нещо повече времевата сложност на $InsertSort$ е $\Theta(n + |\sigma(A)|)$, където:*

$$\sigma(A) = \{(i, j) \,|\, (A[i] - A[j])(i - j) < 0\}.$$

Доказателство. Нека A_ℓ е масивът A преди изпълнението на ред @1 при параметър ℓ. Ще покажем, че:

$$I(\ell) : A_\ell[1..\ell-1] \text{ е сортирана пермутация на } A[1..\ell-1] \text{ и } A_\ell[\ell..n] = A[\ell..n].$$

Доказваме, че $I(\ell)$ е вярно за всяко $\ell \leq n + 1$.

- $\ell = 1$. Тогава $A_1 = A$.
- Нека $I(\ell)$ е вярно и $\ell \le n$. Тогава $A_{\ell+1}$ е модификацията, която претърпява A_ℓ в следствие на $Insert(A_\ell, \ell)$. От предишната лема, това означава, че:

$$A_{\ell+1}[1..\ell] \text{ е сортирана пермутация на } A_\ell[1..\ell] \text{ и } A_{\ell+1}[\ell+1..n] = A_\ell[\ell+1..n] \stackrel{I(\ell)}{=} A[\ell+1..n].$$

Накрая, тъй като $A_\ell[1..\ell-1] \circ A[\ell]$ е пермутация на $A[1..\ell]$, то и $A_{\ell+1}[1..\ell]$ е сортирана пермутация на $A[1..\ell]$.

Остана да анилизраме времевата сложност на алгоритъма. Тъй като $A_\ell[1..\ell-1]$ е пермутация на $A[1..\ell-1]$ и $A_\ell[\ell] = A[\ell]$, то:

$$\sigma^-(A_\ell, \ell) = \sigma^-(A, \ell).$$

Така от предишната лема, получаваме, че времевата сложност на $InsertSort(A)$ е:

$$\sum_{\ell=1}^{n} \Theta(1 + |\sigma^-(A_\ell, \ell)|) = \Theta\left(\sum_{\ell=1}^{n}(1 + |\sigma^-(A_\ell, \ell)|)\right) = \Theta\left(n + \sum_{\ell=1}^{n}|\sigma^-(A, \ell)|\right).$$

Остана да забележим, че $(i,j) \in \sigma(A)$ точно тогава, когато $(i-j)(A[i]-A[j]) < 0$, което е еквивалентно на $(j,i) \in \sigma(A)$ и $i \neq j$. Оттук получаваме, че:

$$|\sigma(A)| = 2|\{(i,j) \mid i < j \& A[j] < A[i]\}| = 2|\bigcup_j \sigma^-(A,j)| = 2\sum_{j=1}^{n}|\sigma^-(A,j)|.$$

С това показахме, че сложността на $InsertSort$ е $\Theta\left(n + \sum_{\ell=1}^{n} |\sigma^-(A,\ell)|\right) = \Theta(n + \frac{1}{2}|\sigma(A)| = \Theta(n + |\sigma(A)|)$. □

Следствие 0.1. *Сложността на InsertSort е $O(n^2)$.*

Доказателство. $|\sigma(A)| \le |\{(i,j) \mid 1 \le i,j \le n\}| = n^2$, откъдето $\Theta(|\sigma(A)|) \subseteq O(n^2)$. □