```
1 for i = 0 to n − 2 do
2     min ← i;
3     for j = i + 1 to n − 1 do
4         if A[j] < A[min] then
5             min ← j;
6         end
7     end
8     swap A[i] and A[min];
9 end
```

Algorithm 5: SelectionSort($A[0..n-1]$)

1. A correção:

(a) Invariante para o laço interno:

> Antes de cada iteração do laço FOR (linhas 3-7), A[min] é o menor elemento do subvetor A[i..j-1].

Inicialização:

Antes da primeira iteração, temos $j = i+1$ e $i = min$. Portanto, A[min] é o menor elemento do subvetor $A[i] = A[min]$.

Manutenção:

Considere a K-ésima ($j = i + K$) iteração. Estamos assumindo que A[min] é o menor elemento do subvetor A[i..i+K-1]. Durante esta iteração, o elemento $A[j] = A[i+K]$ é comparado com o elemento A[min] (linha 4). Temos 2 casos possíveis:

1. $A[i+K] < A[min]$: Por hipótese temos que A[min] é menor ou igual a todo elemento do subvetor A[i..i+K-1]. Como A[i+K] é menor do que A[min], concluímos que A[i+K] é o menor elemento do subvetor A[i..i+K], e como A[i+K] passa a ser A[min], concluímos que A[min] é o menor elemento do subvetor A[i..i+K] de forma que a invariante está satisfeita.

2. $A[i+K] \geq A[min]$: Neste caso, também temos por hipótese que A[min] é menor ou igual a todo elemento do subvetor A[i..i+K-1] e, como adicionalmente $A[min] \leq A[i+K]$ concluímos que A[min] é o menor elemento do subvetor A[i..i+K].

Finalização: Ao final da execução do laço, teremos que $A[min]$ é o menor elemento do subvetor $A[i..n]$.

(b) Invariante para o laço externo (linhas 1-9):

Antes da i-ésima iteração do laço FOR (linhas 1-9), o subvetor $A[0..i-1]$ está ordenado com os i menores elementos de A.

Inicialização: Antes da primeira iteração temos $i=0$ e a invariante vale por vacuidade já que o subvetor $A[0..i-1]$ é vazio.

Manutenção: Agora assumimos que antes da k-ésima iteração do laço FOR ($i=k-1$), o subvetor $A[0..k-2]$ está ordenado com os $(k-1)$ menores elementos de A. Durante a execução deste laço, os elementos $A[min]$ e $A[i]$ são permutados, e pela invariante do laço interno, sabemos que $A[min]$ é o menor elemento do subvetor $A[k-1..n]$. Portanto, ao final desta iteração o subvetor $A[0..k-1]$ está ordenado com os k menores elementos de A.

Finalização: Ao final da execução do laço externo, temos $i=n$, e a invariante nos diz que o subvetor $A[0..n-1]$ está ordenado com os n menores elementos do vetor A. Ou seja, SelectionSort é correto ∎

② Observe que este algoritmo executa sempre o mesmo número de passos, e portanto a análise do pior caso coincide com a análise do melhor caso. Temos:

$$T(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} (n-1-(i+1)+1) = \sum_{i=0}^{n-2} (n-i-1) = \sum_{i=1}^{n-1} i = \Theta(n^2).$$

∎