Selection Sort

$\text{array} = [\overset{19}{\cancel{50}}, 38, 45, 79, \overset{50}{\cancel{19}}, 27, 29]$

0 1 2 3 4 5 6

min = ~~0~~ ~~1~~ 4

In Pass 0 start by considering that min index is 0 and keep changing this index with the index of next smaller value further obtained in the process of traversing through the list. At last when whole list is traversed swap the index 0 with min index value if found. Perform the same with start index 1 for Pass 1 and similarly continue for other Passes as well.

Pass 1

↳ index of the minimum value

$[19, (\overset{27}{\cancel{38}}, 45, 79, 50, \overset{38}{\cancel{27}}, 29)]$

0 1 2 3 4 5 6

min = ~~1~~ 5

Pass 2

$[19, 27, (\overset{29}{\cancel{45}}, 79, 50, 38, \overset{45}{\cancel{29}})]$

0 1 2 3 4 5 6

min = ~~2~~ ~~5~~ 6

Pass3

$[19, 27, 29, (\overset{38}{\cancel{79}}, 50, \overset{79}{\cancel{38}}, 45)]$

0 1 2 3 4 5 6

min = ~~3~~ ~~4~~ 5

Pass4

$19, 27, 29, 38, (\overset{45}{\cancel{50}}, 79, \overset{50}{\cancel{45}})$

0 1 2 3 4 5 6

min = ~~4~~ 6

Pass5

$19, 27, 29, 38, 45, (\overset{50}{\cancel{79}}, \overset{79}{\cancel{50}})$

0 1 2 3 4 5 6

min = ~~5~~ 6

Pass6

0 1 2 3 4 5 6

19, 27, 29, 38, 45, 50, 79

Note

↳ At every pass, only one swap is required.

Because of this reason it's slightly better as compared to Bubble sort where we were performing swapping for each consecutive elements if the next element is smaller than the previous element.

swaps

$$\left.\begin{array}{l} (n-1)\text{th Pass} \rightarrow 1 \text{ swap} \\ \quad\vdots \\ 1\text{st Pass} \rightarrow 1 \text{ swap} \end{array}\right\} \quad (n-1) \text{ swaps} \hookrightarrow O(n)$$

n-1 total swaps will be required.

Time complexity wrt the number of swaps is O(n) which in case of Bubble sort was O(n*n) or O(n sqaure)

comparisons

Time complexity wrt no. of comparisons

$$\hookrightarrow \underbrace{(n-1)}_{\text{no. of comparisons in 1st pass}} + \underbrace{(n-2)}_{\text{no. of comparisons in 2nd pass}} + (n-3) + \dots + \underbrace{1}_{\text{no. of comparison s in last pass}} = \frac{(n-1)n}{2}$$

$$\Rightarrow O(n^2)$$

Time complexity for selection sort wrt to comparisons is O(nsqaure) which is equivalent to what we were having in Bubble sort wrt to comparisons

$$\underline{\text{Time complexity}} \rightarrow \left.\begin{cases} \text{comparisons} \rightarrow O(n^2) \\ \text{swaps} \rightarrow O(n) \end{cases}\right\} \Rightarrow O(n^2)$$

Over all time complexity for Selection sort.

We can conclude that Selection sort works better as compared to Bubble sort wrt to no. of swaps. However, overall time complexity is of order n square.

Please note that in Selection sort we are sorting elements from left extreme end that is we are generating the smallest element that will appear in the sorted list first via each pass.
Whereas, in Bubble sort we are sorting elements from right extreme end that is we are generating the largest element that will appear in the sorted list first via each pass.