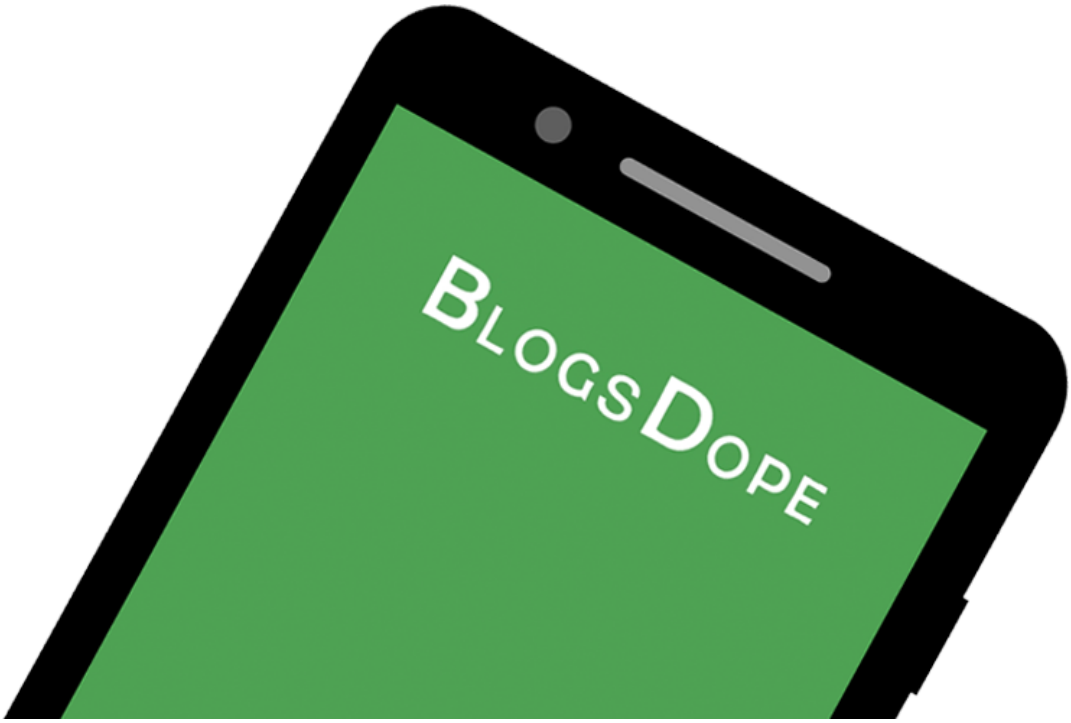
# Sorting a list using selection sort in Python

May 19, 2017 PYTHON (/blog/tag/python/?tag=python) ARRAY (/blog/tag/array/?tag=array) SORT (/blog/tag/sort/?tag=sort) ALGORITHM (/blog/tag/algorithm/?tag=algorithm) LOOP (/blog/tag/loop/?tag=loop)
17608





(https://play.google.com/store/apps/details?id=com.blogsdope&pcampaignid=MKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1)



Selection sort is one of the simplest sorting algorithms. It is similar to the hand picking where we take the smallest element and put it in the first position and the second smallest at the second position and so on. It is also similar. We first check for smallest element in the list and swap it with the first element of the list. Again, we check for the smallest number in a sublist, excluding the first element of the list as it is where it should be (at the first position) and put it in the second position of the list. We continue repeating this process until the list gets sorted.

The time complexity of selection sort is ($O(n^2)$).

We follow the following steps to perform selection sort:

1. Start from the first element in the list and search for the smallest element in the list.
2. Swap the first element with the smallest element of the list.
3. Take a sublist (excluding the first element of the list as it is at its place) and search for the smallest number in the sublist (second smallest number of the entire list) and swap it with the first element of the list (second element of the entire list).
4. Repeat the steps 2 and 3 with new sublists until the list gets sorted.

## Working of selection sort:

Initial list

| | | | | | | |
|---|---|---|---|---|---|---|
| 16 | 19 | 11 | 15 | 10 | 12 | 14 |

First iteration

| | | | | | | |
|---|---|---|---|---|---|---|
| 16 | 19 | 11 | 15 | 10 | 12 | 14 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 10 | 19 | 11 | 15 | 16 | 12 | 14 |

Second iteration

| | | | | | | |
|---|---|---|---|---|---|---|
| 10 | 19 | 11 | 15 | 16 | 12 | 14 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 10 | 11 | 19 | 15 | 16 | 12 | 14 |

Third iteration

| | | | | | | |
|---|---|---|---|---|---|---|
| 10 | 11 | 19 | 15 | 16 | 12 | 14 |

| 10 | 11 | 12 | 15 | 16 | 19 | 14 |
|---|---|---|---|---|---|---|

Fourth iteration

| 10 | 11 | 12 | 15 | 16 | 19 | 14 |
|---|---|---|---|---|---|---|

| 10 | 11 | 12 | 14 | 16 | 19 | 15 |
|---|---|---|---|---|---|---|

Fifth iteration

| 10 | 11 | 12 | 14 | 16 | 19 | 15 |
|---|---|---|---|---|---|---|

| 10 | 11 | 12 | 14 | 15 | 19 | 16 |
|---|---|---|---|---|---|---|

Sixth iteration

| 10 | 11 | 12 | 14 | 15 | 19 | 16 |
|---|---|---|---|---|---|---|

| 10 | 11 | 12 | 14 | 15 | 16 | 19 |
|---|---|---|---|---|---|---|

Final list

| 10 | 11 | 12 | 14 | 15 | 16 | 19 |
|---|---|---|---|---|---|---|

Let’s code this in Python

```
a = [16, 19, 11, 15, 10, 12, 14]

i = 0
while i<len(a):
    #smallest element in the sublist
    smallest = min(a[i:])
    #index of smallest element
    index_of_smallest = a.index(smallest)
    #swapping
    a[i],a[index_of_smallest] = a[index_of_smallest],a[i]
    i=i+1
print (a)
```

The code is just an implementation of the above explanation. We are finding the smallest number by using the 'min' function. and then swapping it with the first element of the sublist. You can see this (https://www.codesdope.com/blog/article/functions-we-can-use-on-a-pythons-list/) article for the list of the functions available on a Python's list.



## Liked the post?

(https://www.facebook.com/sharer/sharer.php?u=https://www.codesdope.com/blog/article/sorting-a-list-using-selection-sort-in-python/) (https://twitter.com/intent/tweet?url=https://www.codesdope.com/blog/article/sorting-a-list-using-selection-sort-in-python/&text=Sorting a list using selection sort in Python &via=codesdope) (https://plus.google.com/share?url=https://www.codesdope.com/blog/article/sorting-a-list-using-selection-sort-in-python/) (https://www.linkedin.com/shareArticle?url=https://www.codesdope.com/blog/article/sorting-a-list-using-selection-