

# Sorting Algorithms: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2021

## Syntax

- Swapping the values of two indexes in a list:

```
def swap(values, i, j):  
    temp = values[i]  
    values[i] = values[j]  
    values[j] = temp
```

- Selection sort:

```
def select_minimum_index_in_range(values, range_start):  
    minimum = None  
    minimum_index = None  
    N = len(values)  
    for i in range(range_start, N):  
        if minimum == None or values[i] < minimum:  
            minimum = values[i]  
            minimum_index = i  
    return minimum_index  
  
def selection_sort(values):  
    N = len(values)  
    for range_start in range(N):  
        index = select_minimum_index_in_range(values, range_start)  
        swap(values, range_start, index)
```

## Concepts

- The sum of the first  $N$  natural numbers is equal to  $N^2 / 2 + N / 2$  which is  $O(N^2)$ .
- Optimizing a double `for` loop to only consider distinct pairs will run faster in practice but will not improve the time complexity of an algorithm.
- Python is able to sort list very quickly in  $O(N \log(N))$ .

## Resources

- [Selection sort](#)
- [Sorting algorithms](#)