Selection Sort

Vivek K. S., Deepak G.

Information Systems Decision Sciences (ISDS)
MUMA College of Business
University of South Florida
Tampa, Florida

2017

Introduction

- The selection sort improves on the bubble sort by making only one exchange for every pass through the list.
- As every pass is made, the largest number is identified and placed in the proper position.
- This process takes n-1 passes for a list of n items.

Implementation in Python

The Python implementation is as follows:

```
def selection_sort(a_list):
    for fill_slot in range(len(a_list) - 1, 0, -1):
        pos_of_max = 0
        for location in range(1, fill_slot + 1):
            if a_list[location] > a_list[pos_of_max]:
                pos_of_max = location
            a_list[fill_slot],a_list[pos_of_max] =
            a_list[pos_of_max],a_list[fill_slot]
a = [1,2,3,4,6,345,25,6,25,5,6,72,61,6,262]
selection_sort(a)
[1, 2, 3, 4, 5, 6, 6, 6, 6, 25, 25, 61, 72, 262, 345]
```

Analysis of Selection Sort

- Selection sort makes the same number of comparisons as the bubble sort.
- It is therefore also $\mathrm{O}(n^2)$.
- But there is a definite reduction in the number of exchanges and therefore is faster than Bubble sort.

Summary

- Selection sort is yet another simple sorting algorithm which offers a little advantage over the Bubble sort.
- Only one item is sorted in each pass.
- The performance is still O((n^2)).