

LAB211 Assignment

Type:
Code:
LOC:
Slot(s):

Short Assignment
J1.S.P0002
40
1

Title

Selection sort algorithm.

Background Context

Selection sort is one of the $O(n^2)$ sorting algorithms, which makes it quite inefficient for sorting large data volumes. Selection sort is notable for its programming simplicity and it can over perform other sorts in certain situations (see complexity analysis for more details).

Program Specifications

Design a program that allows users to input the number of array. Generate random integer in number range input. Display unsorted array and sorted array using selection sort.

Function details:

1. Display a screen to prompt users to input a positive decimal number.
 - Users run the program, display a screen to ask users to enter a positive decimal number.
 - Users input a positive decimal number. Then, perform **Function 2**.
2. Display & sort array.
 - Generate random integer in number range for each array element.
 - Display array before and after sorting.

Expectation of User interface:

```
Enter number of array:
10
Unsorted array: [2, 6, 3, 6, 8, 6, 1, 2, 9, 8]
Sorted array: [1, 2, 2, 3, 6, 6, 6, 8, 8, 9]BUILD SUCCESSFUL (total time: 1 second)
```

Guidelines

Algorithm

The idea of algorithm is quite simple. Array is imaginary divided into two parts - **sorted one** and **unsorted one**. At the beginning, **sorted part** is **empty**, while **unsorted one** contains **whole array**. At every step, algorithm finds **minimal element** in the **unsorted part** and adds it to the end of the **sorted one**. When **unsorted part** becomes **empty**, algorithm *stops*.

When algorithm sorts an array, it swaps first element of unsorted part with minimal element and then it is included to the sorted part. This implementation of selection sort is **not stable**. In case of linked list is sorted, and, instead of swaps, minimal element is linked to the unsorted part, selection sort is **stable**.

Let us see an example of sorting an array to make the idea of selection sort clearer.

Example. Sort {5, 1, 12, -5, 16, 2, 12, 14} using selection sort.

5 1 12 -5 16 2 12 14

5 1 12 -5 16 2 12 14



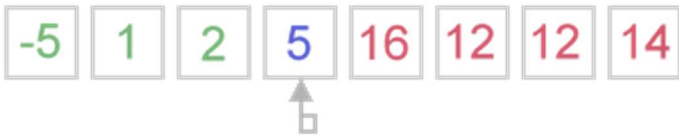
-5 1 12 5 16 2 12 14



-5 1 12 5 16 2 12 14



-5 1 2 5 16 12 12 14



-5 1 2 5 16 12 12 14



-5 1 2 5 12 16 12 14



-5 1 2 5 12 12 16 14



-5 1 2 5 12 12 14 16