



Data Structure

Lab Session #8: Internal Sorting

U Kang
Seoul National University



Goals

- Implement “**Stable Selection Sort**”
 - We recommend you to implement with following steps
 - `print()`
 - `setItems()`
 - `check()`
 - `doStableSelectionSort()`
- Optional “Sorting Sequence visualization”



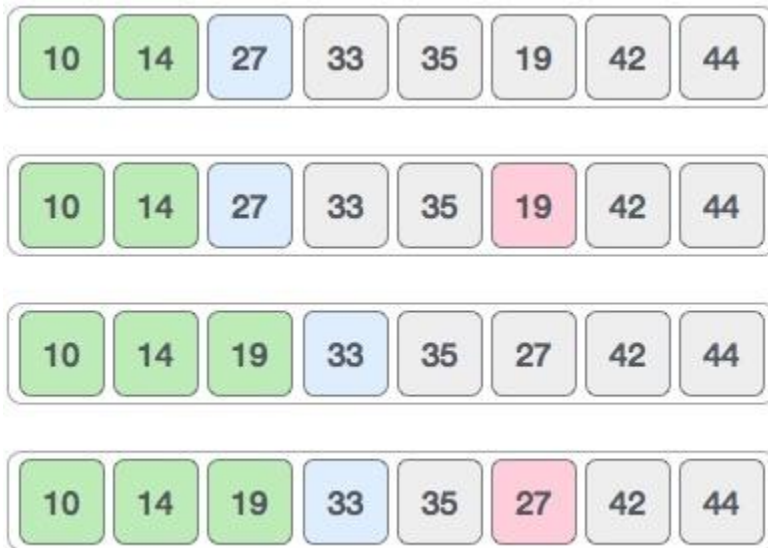
Build a project

- Download the project for this lab from eTL.
- Extract the project, and import it using IntelliJ



Selection sort

- The sorting algorithm that sorts an array by repeatedly finding the minimum element from unsorted part and putting it at the beginning.





Stable sorting algorithm

- A sorting algorithm is said to be stable if two objects with equal keys appear in the same order in sorted output as they appear in the input array to be sorted.

Input data	3 (A)	2 (B)	2 (C)	6 (D)	7 (F)
Sorted but not stable	2 (C)	2 (B)	3 (A)	6 (D)	7 (F)
Sorted and stable	2 (B)	2 (C)	3 (A)	6 (D)	7 (F)



I/O Specification

■ print

Input format	Output format
<code>print</code>	
Description	
<ul style="list-style-type: none">- Print the list of nodes in the 'this.items'.- You should use 'Node.toString()' when you print key and item of the nodes.	
Example Input	Example Output
<code>print()</code>	<code>[1 A] [2 B]</code>



I/O Specification

■ setItems

Input format	Output format
setItems (n) (itemLine)	
Description	
<ul style="list-style-type: none">- Set items in the itemLine to the 'this.items'.- Print list of items by calling 'print' function at the end.	
Example Input	Example Output
SET_ITEMS 2	[1 A] [2 B]



I/O Specification

■ check

Input format	Output format
check	
Description	
<ul style="list-style-type: none">- Check if the SortManager has the sorted and stable items.- Print “Unsorted” if ‘this.items’ are not sorted.- Print “Sorted, but not stable” if “this.items” are sorted but not stable.- Print “Sorted and stable” if “this.items” are both sorted and stable.- Assume, all items are given lexicographical order at first and there are no duplicate items.	
Example Input	Example Output
CHECK	Sorted and stable



I/O Specification

■ doStableSelectionSort

Input format	Output format
doStableSelectionSort	
Description	
<ul style="list-style-type: none">- Sort the items in the “this.items” by <u>comparing key</u>.- You have to implement stable selection sorting algorithm.- Print list of items by calling ‘print’ function at the end.- Note that we cannot use the node’s item when we sort “this.items.”	
Example Input	Example Output
STABLE_SORT	[1 A] [2 B]



Sample Input & Output

<Input>

```
SET_ITEMS 3
3_A 2_B 1_C
CHECK
SET_ITEMS 4
1_A 2_C 2_B 3_D
CHECK
SET_ITEMS 4
1_A 2_B 2_C 4_D
CHECK
SET_ITEMS 10
50_A 15_B 15_C 1_D 10_E 36_F 1
5_G 1_H 3_I 37_J
STABLE_SORT
CHECK
```

<Output>

```
SET_ITEMS 3
[ 3 A] [ 2 B] [ 1 C]
CHECK
Unsorted.
SET_ITEMS 4
[ 1 A] [ 2 C] [ 2 B] [ 3 D]
CHECK
Sorted, but not stable.
SET_ITEMS 4
[ 1 A] [ 2 B] [ 2 C] [ 4 D]
CHECK
Sorted and stable.
SET_ITEMS 10
[50 A] [15 B] [15 C] [ 1 D] [10 E] [36 F] [15 G] [ 1 H] [ 3 I] [37 J]
STABLE_SORT
[ 1 D] [ 1 H] [ 3 I] [10 E] [15 B] [15 C] [15 G] [36 F] [37 J] [50 A]
CHECK
Sorted and stable.
```

* These files are in the testcase folder!



Sorting Sequence visualization

- This problem is optional, you don't have to check this problem to T.A..
- Can you visualize the sequence of sorting by printing them?
 - Print format
 - For unsorted item
 - (Item)/ [][][][]...[] (number of boxes = key value)
 - For sorted item
 - (Item)/ [*][*][*][*]...[*] (number of boxes = key value)



Sorting Sequence visualization

- Input: SHOW
- Print

```
A/ [ ] [ ]  
B/ [ ] [ ] [ ] [ ] [ ]  
C/ [ ] [ ]  
D/ [ ] [ ] [ ] [ ] [ ] [ ]  
E/ [ ]
```

```
E/ [*]  
B/ [ ] [ ] [ ] [ ] [ ]  
C/ [ ] [ ]  
D/ [ ] [ ] [ ] [ ] [ ] [ ]  
A/ [ ] [ ]
```

```
E/ [*]  
A/ [*] [*]  
C/ [ ] [ ]  
D/ [ ] [ ] [ ] [ ] [ ] [ ]  
B/ [ ] [ ] [ ] [ ] [ ]
```

```
E/ [*]  
A/ [*] [*]  
C/ [*] [*]  
D/ [ ] [ ] [ ] [ ] [ ] [ ]  
B/ [ ] [ ] [ ] [ ] [ ]
```

```
E/ [*]  
A/ [*] [*]  
C/ [*] [*]  
B/ [*] [*] [*] [*] [*]  
D/ [ ] [ ] [ ] [ ] [ ] [ ]
```



```
E/ [*]  
A/ [*] [*]  
C/ [*] [*]  
B/ [*] [*] [*] [*] [*]  
D/ [*] [*] [*] [*] [*] [*]
```



Questions?