Data Structures (Spring 2020)

Sorting Algorithms (12th Lab)

2020.06.05

Seoul National University
Database Systems Lab

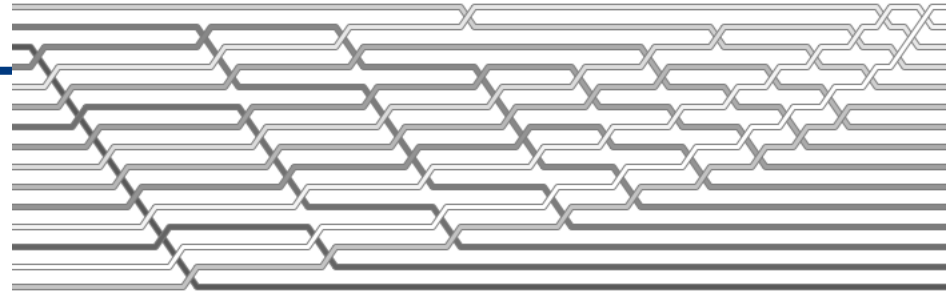
Today's Lab

- Sorting Algorithm
  - Bubble Sort
  - Selection Sort
  - Insertion Sort

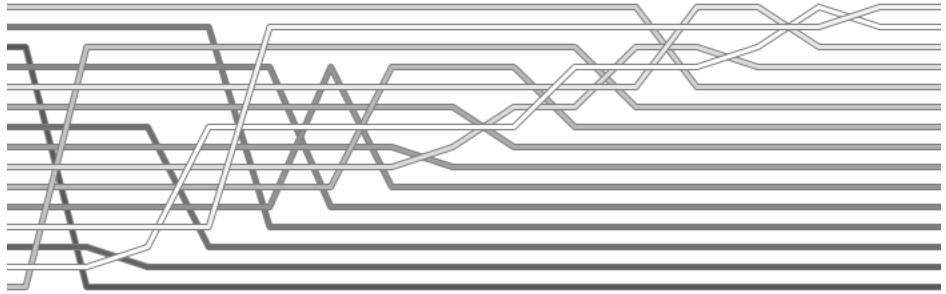
Sorting Algorithm

- Sorting Algorithm
  - An algorithm that puts elements of a list in a certain order.
  - The most frequently used orders are numerical order and lexicographical order.

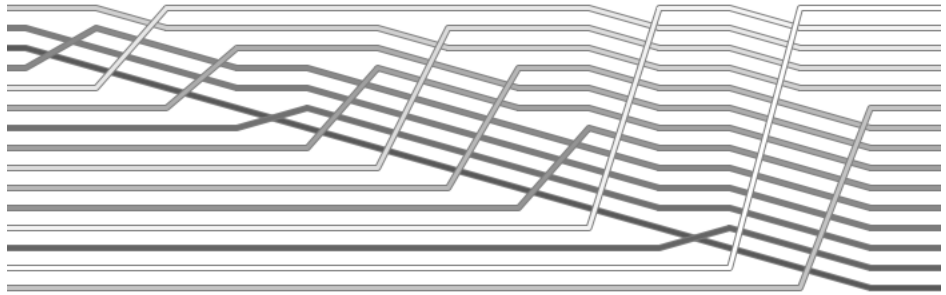
- Bubble sort: larger (or smaller) value float to surface (**like bubbles!**).
- Selection sort: **select minimum value** and swap with head.
- Insertion sort: **insert** an element **into sorted list**. (similar to how we put money in our wallet!)



bubble sort visualization



selection sort visualization



insertion sort visualization

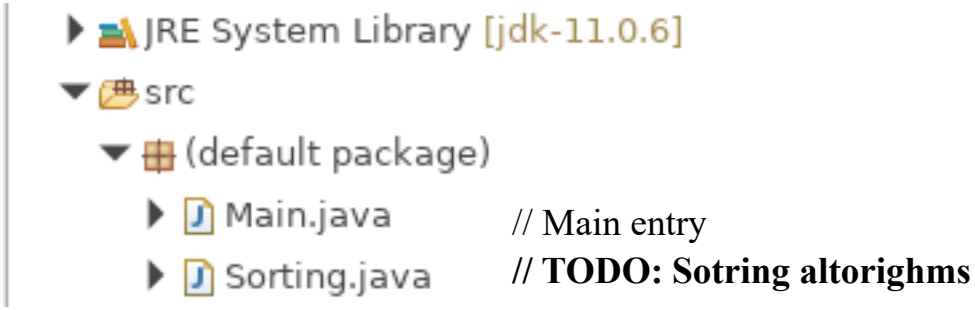
"Sorting algorithm", https://en.wikipedia.org/wiki/Sorting_algorithm
Cortesi, Aldo (27 April 2007). "Visualising Sorting Algorithms". https://corte.si/posts/code/visualisingsorting/index.html, Retrieved 5 Jun 2020.

Sorting Algorithm spec

- All output should be ascending order.
- `public static void BubbleSort(Integer[] data);`
  - Integer[] data: an input array and an output sorted array
- `public static void SelectionSort(Integer[] data);`
  - Integer[] data: an input array and an output sorted array
- `public static void InsertionSort(Integer[] data);`
  - Integer[] data: an input array and an output sorted array

Exercises

- Fill the blank of codes
  - Update your code in `Sorting.java` (`"// TODO: "` section)
  - Write `BubbleSort(), InsertionSort(), SelectionSort()` method



Project Structure

```java
public static void main(String[] args) {
    // initialize
    int size = Integer.valueOf(args[0]);
    int[] ref = Stream.iterate(0, x -> x + 1).limit(size).mapToInt(Integer::intValue).toArray();
    int[] array = new int[size];

    // shuffle
    naiveShuffle(ref);

    // initial copy
    System.arraycopy(ref, 0, array, 0, size);
    System.out.print("Initial array\t: "); printArray(array);

    // Bubble sort
    System.arraycopy(ref, 0, array, 0, size);
    Sorting.BubbleSort(array);
    System.out.print("Bubble Sort\t: "); printArray(array);

    // selection sort
    System.arraycopy(ref, 0, array, 0, size);
    Sorting.SelectionSort(array);
    System.out.print("Selection Sort\t: "); printArray(array);

    // Insertion sort
    System.arraycopy(ref, 0, array, 0, size);
    Sorting.InsertionSort(array);
    System.out.print("Insertion Sort\t: "); printArray(array);
}
```

Main.java

Exercises

- Result

```
$ java Main 16
Initial array    : 15,10,2,9,4,6,14,1,8,5,13,12,0,11,7,3,
Bubble Sort      : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,
Selection Sort   : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,
Insertion Sort   : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,

$ java Main 16
Initial array    : 5,15,2,11,12,9,7,6,1,3,10,8,0,14,4,13,
Bubble Sort      : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,
Selection Sort   : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,
Insertion Sort   : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,
```