1 Sorting (Part 1)

1.1 Bubble sort

Bubble sort is the sorting method that the selected value will compare and swap location in every pair until reach the last value of array. This meant this method will slow at start pace but faster after sorting.

Figure 1: Example array

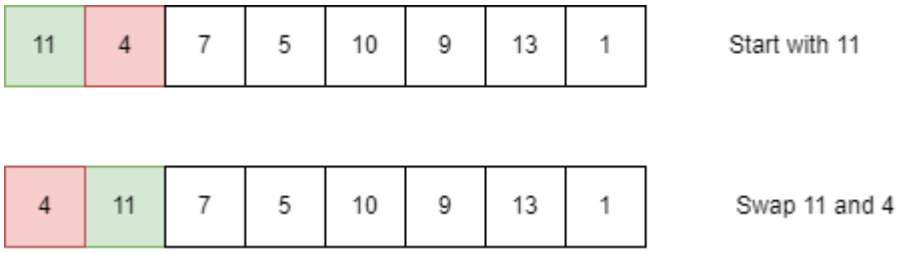

Figure 2: Paring with 1st index and 2nd index then swap

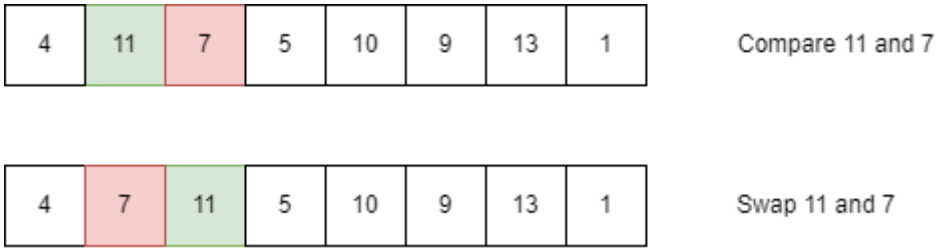

Figure 3: Paring with 2nd index and 3rd index then swap

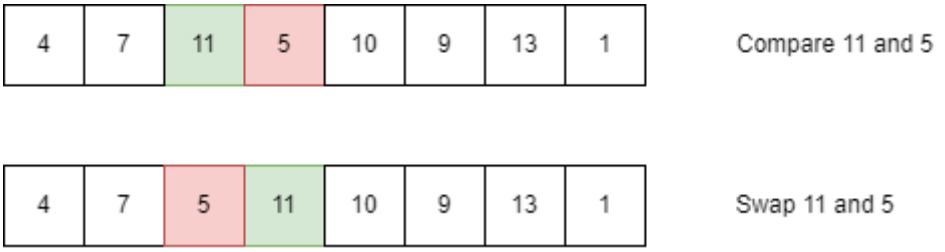

Figure 4: Paring with 3rd index and 4th index then swap

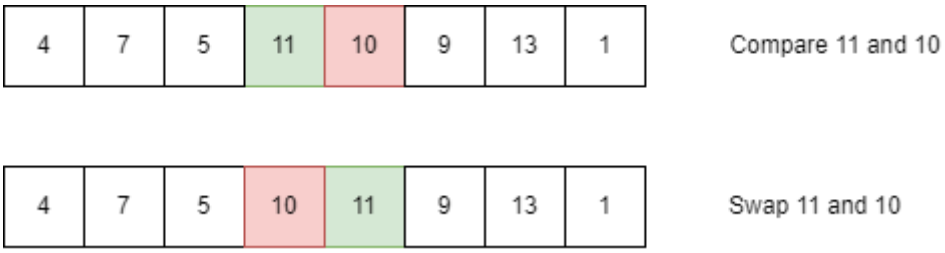

Figure 5: Paring with 4th index and 5th index then swap

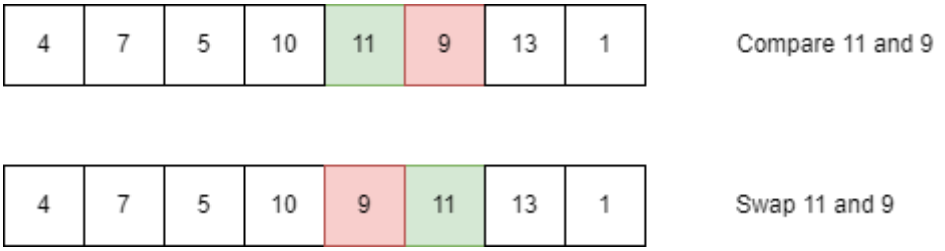

Figure 6: Paring with 5th index and 6th index then swap

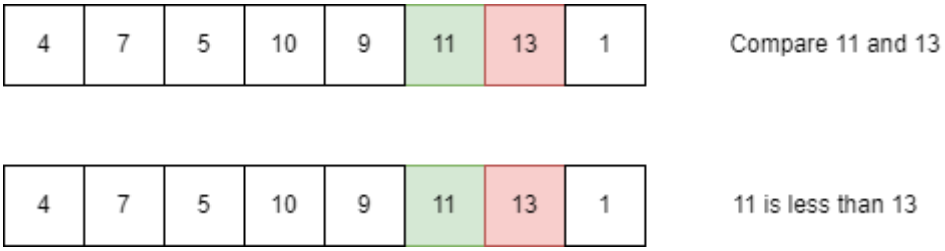

Figure 7: Paring with 6th index and 7th index but 6th index value is less than 7th index value

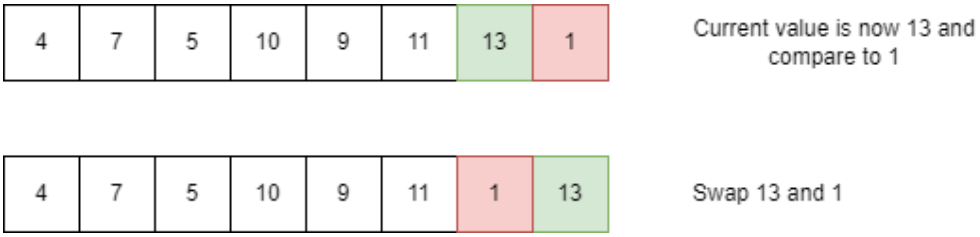

Figure 8: Current value is now index 7th because 7th index value is more than 6th index value then compare and swap value between 7th index and 8th index

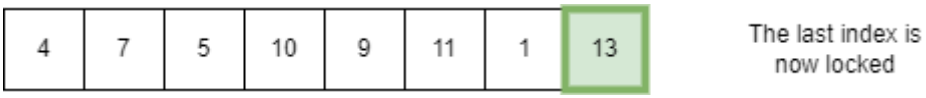

Figure 9: The last index is now locked.

This mean in every run, the locked index will greater and the sorting will be faster.

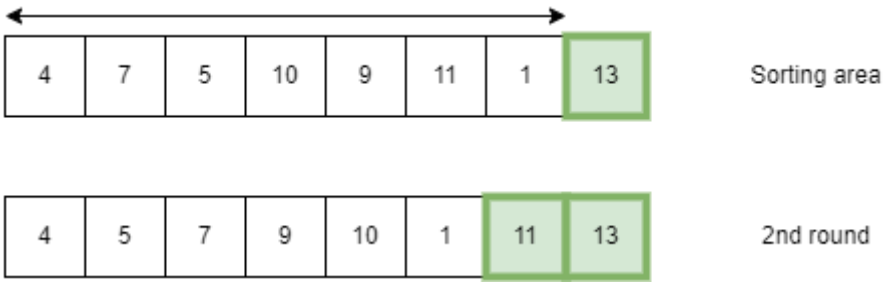

Figure 10: 2nd run with remaining remaing area/index

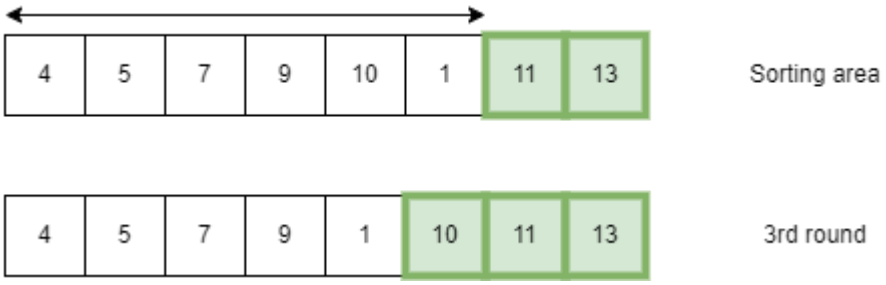

Figure 11: 3rd run with remaining remaing area/index

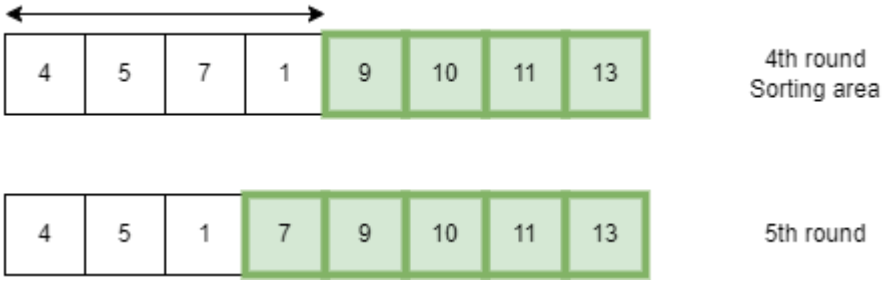

Figure 12: 5th run with remaining remaing area/index

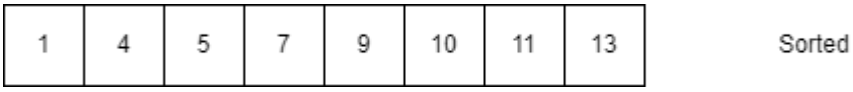

Figure 13: Sorted array

1.2 Selection sort

Selection sort is the method to find the lowest value from array then swap to the index of current run. The first run index will equal to 0 and the third run index will equal to 2 which means index will equal running round - 1.

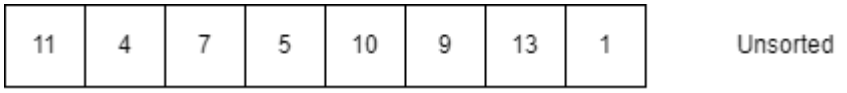

Figure 14: Example array

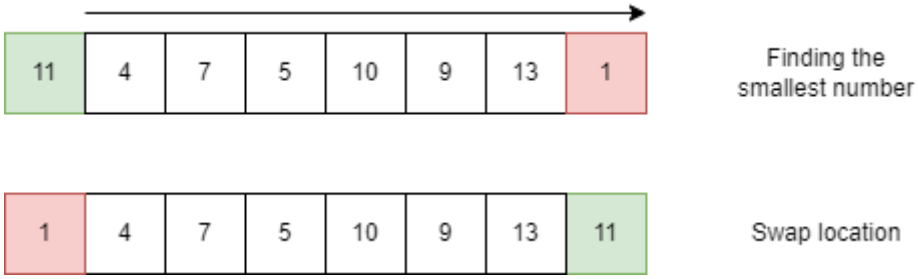

Figure 14: Find the minimum value and swap to the 1st index as the first run.

Figure 15: Lock 1st index after swapped.

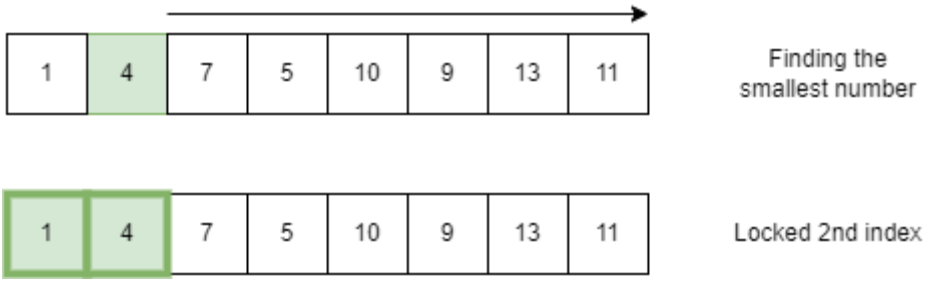

Figure 16: Find the minimum value after 2nd index but its value is the smallest

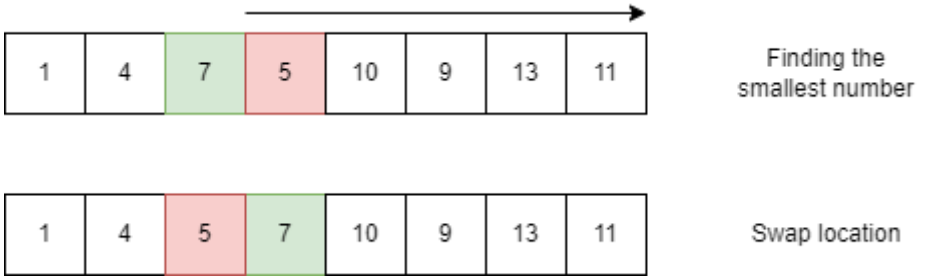

Figure 17: Find the minimum value and swap to the 3rd index as the third run.

Figure 18: Lock 3rd index after swapped.

With this method (Selection sort), the performance is better than bubble sort method because this method does not need to compare and swap every value when during searching. This method needs only find the minimum value of the remaining index and swap to the address of the current run which equal to.

Swapped index = running round - 1.

1.3 Insertion sort

Insertion sort is similar to bubble sort but only compare and will stop when current value is less than the previous value. After insert sorted value the new run will started.

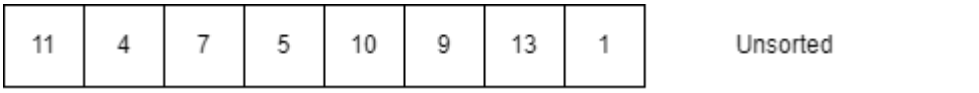

Figure 19: Example array

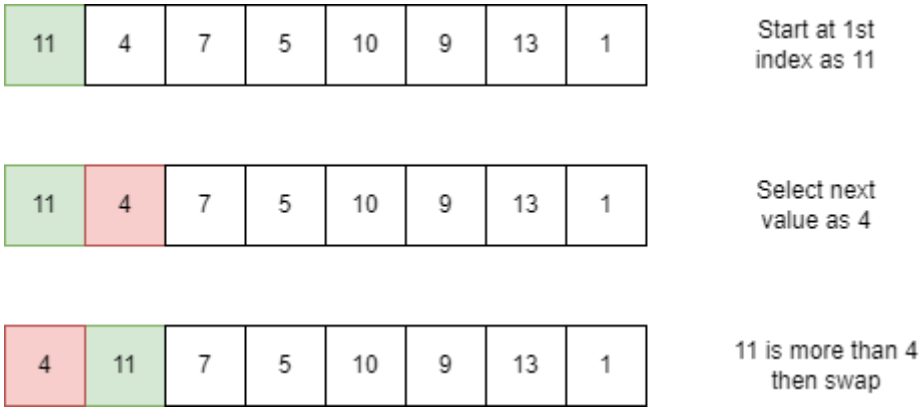

Figure 20: Insertion sort 1st run

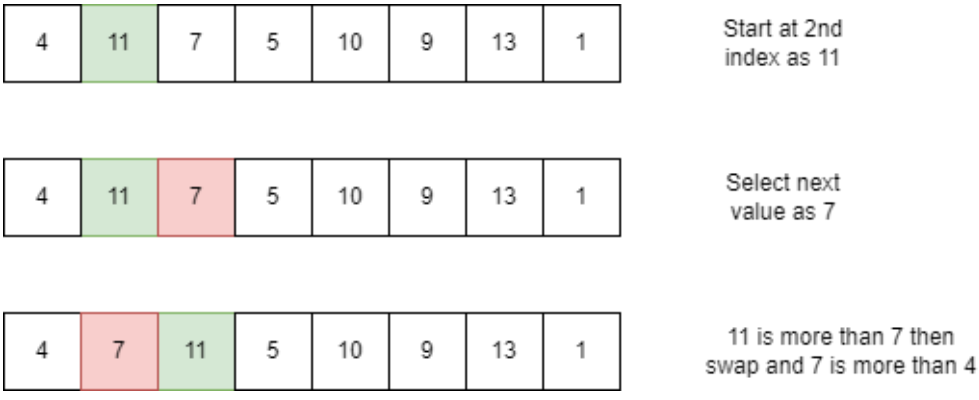

Figure 21: Insertion sort 2nd run

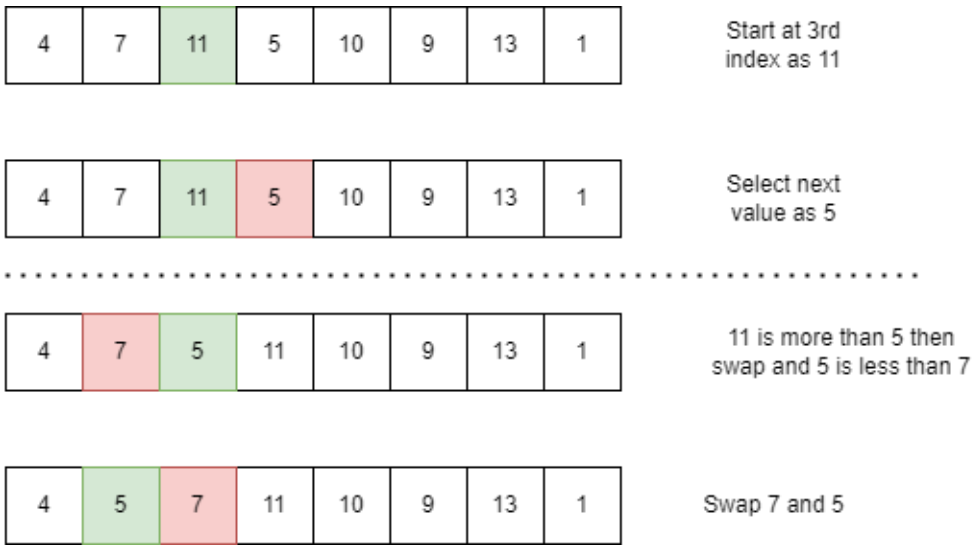

Figure 22: Insertion sort 3rd run

However in figure 22, the step is a bit longer because the selected value (5) is less than the previous value (7). Then the selected value (5) will swap with previous value until selected value (5) is more than the previous value (4).

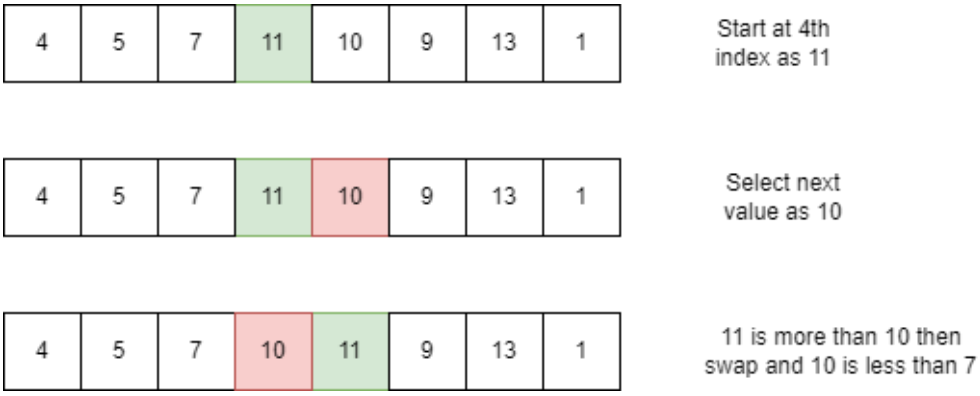

Figure 23: Insertion sort 4th run

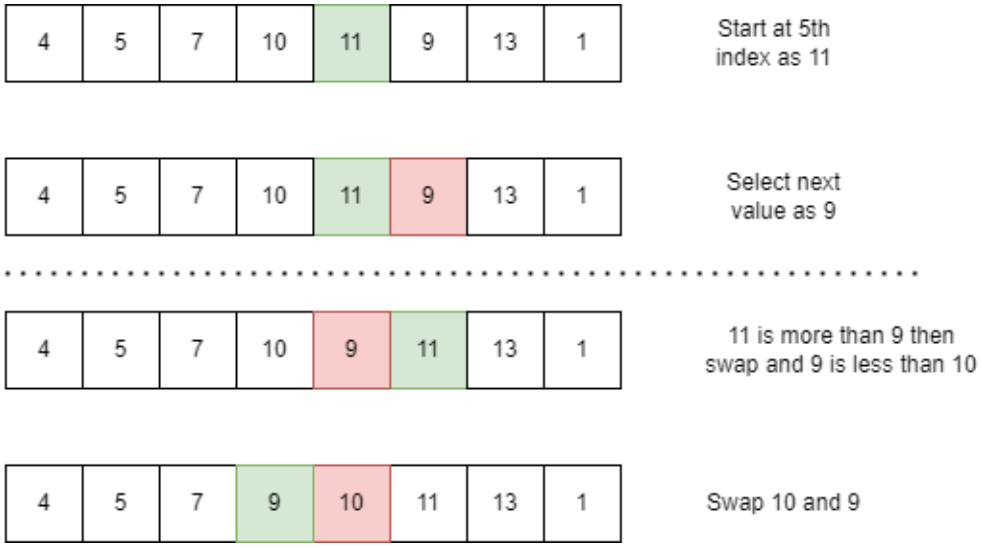

Figure 24: Insertion sort 5th run which similar to the 3rd run

Figure 25: Insertion sort 6th run and no need to swap in this run

Figure 26: Insertion sort 7th run which similar to the 3rd run (part 1)

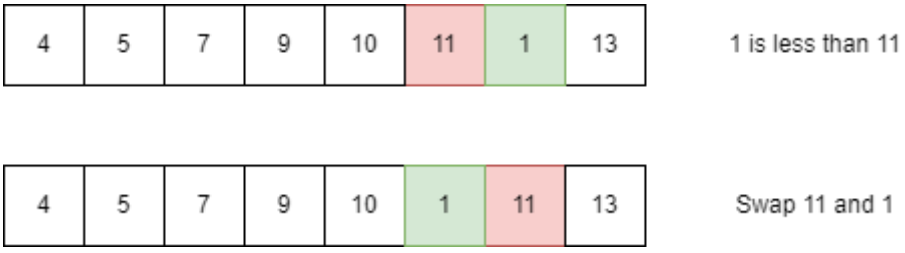

Figure 27: Insertion sort 7th run which similar to the 3rd run (part 2)

Figure 28: Insertion sort 7th run which similar to the 3rd run (part 3)

Figure 29: Insertion sort 7th run which similar to the 3rd run (part 4)

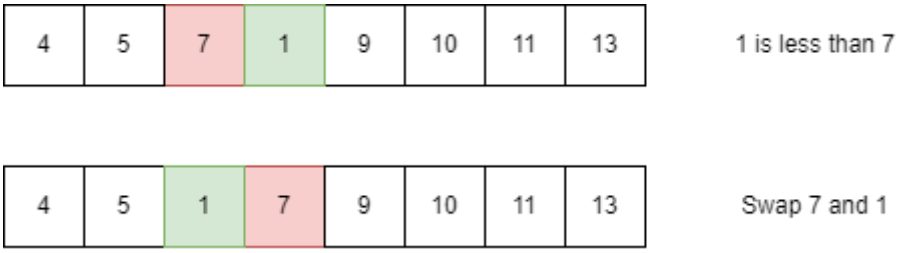

Figure 30: Insertion sort 7th run which similar to the 3rd run (part 5)

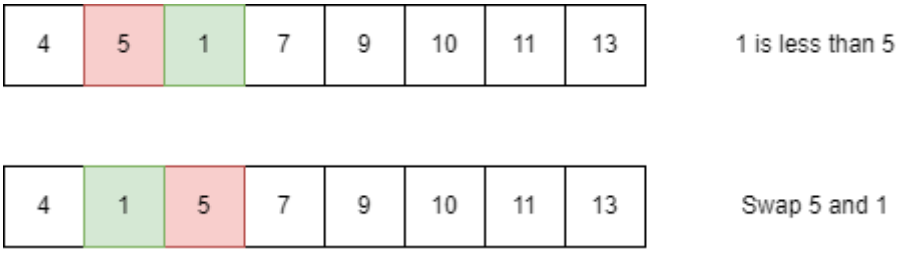

Figure 31: Insertion sort 7th run which similar to the 3rd run (part 6)

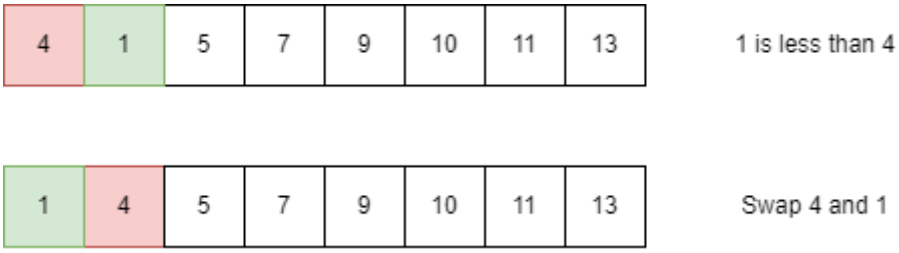

Figure 32: Insertion sort 7th run which similar to the 3rd run (part 7)

As demonstration, every method has their own advantage. It depends on developer conditions and resource management. Because some method applied better performance but required bigger resource.

2 Exercise

2.1 Coding

2.1.1 Requirement

Functional coding

- Bubble sort
- Selection sort
- Insertion sort

2.2 Explanation

2.2.1 From all above method which one is the best in term of performance?

2.2.2 From all above method which one is the best in term of resource management?

2.2.3 From all above method which one is the best in your opinion?