

# Algorithms and data structures

## Tutorial 5: Searching and Sorting Algorithms

Dr. Doan Nhat Quang: doan-nhat.quang@usth.edu.vn

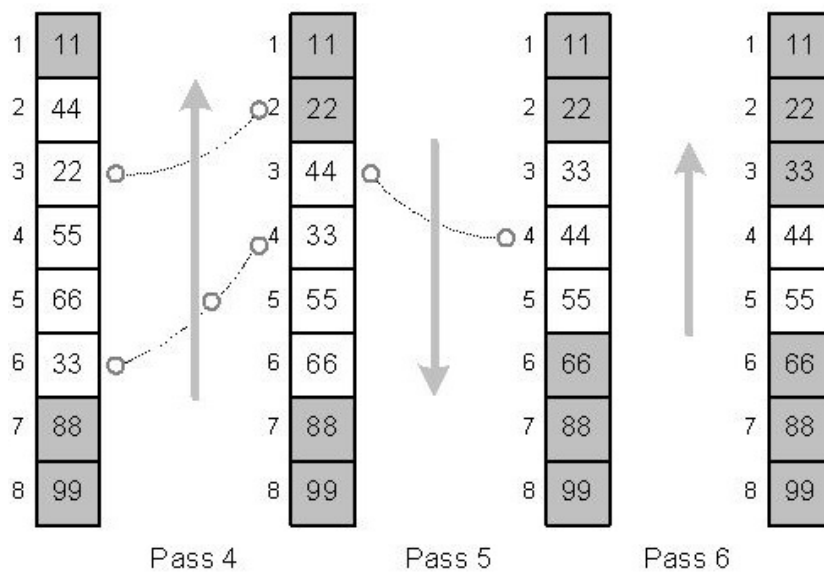
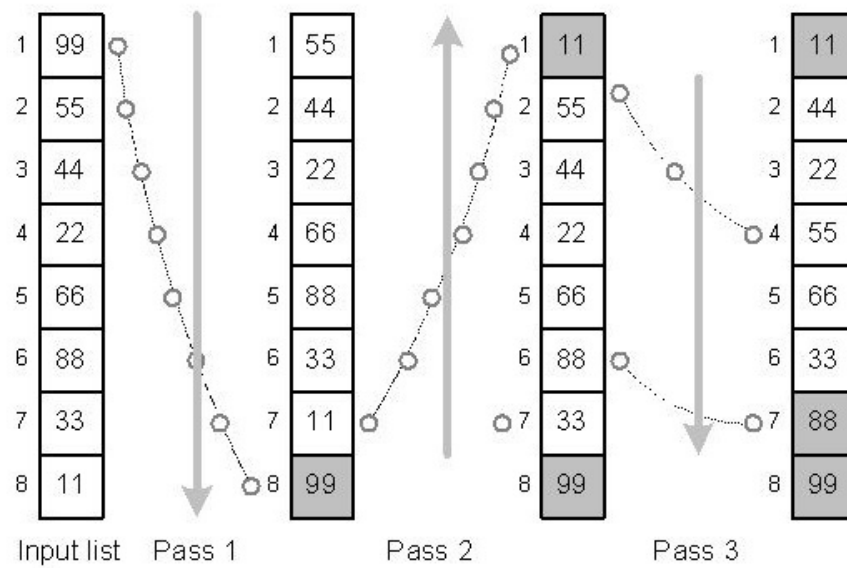
### 1. Searching and Sorting

#### Exercise 1:

In this problem, we would like to implement a variation of the Bubble Sort algorithm. The algorithm differs from a bubble sort in that it sorts in both directions on each pass through the list. The algorithm is illustrated as in the following figure:

- For the first step, we perform bubble sort from the index 1 to  $n$  ( $n$  is the number of elements in the array).
- The next step, we perform a reserved bubble sort from the index  $n$  to 1.
- The process is repeated until all the array is sorted.

Implement and test this algorithm in C/C++. Analyze and compute the complexity of this algorithm in the best, average and worst scenarios.



### Exercise 2:

To avoid doubling the workspace needed when arrays are sorted with Merge Sort, it may be better to use a linked list of data instead of classic arrays (static or dynamic array). Implement Merge Sort in C/C++ and discuss its complexity.