

# LAB 1

LNTHÀNH | FIT@HCMUS | DATA STRUCTURES AND ALGORITHMS

## Objective

In this lab session, we will experiment with number of sorting algorithms.

## Description

### EXERCISE 1-1.

Write sort algorithms include insertion sort, heap sort, quick sort, merge sort for interger array with random values. Compare the system time on a random array of size 10000 or more.

### EXERCISE 1-2.

Sort integer array with even numbers in ascending order and odd numbers in descending order. Notes that, positions of even and odd parts is no change.

For example:

1 2 4 7 15 18 8 9 11

After sorting: 15 2 4 11 9 8 18 7 1

### EXERCISE 1-3.

Read the data from the 'employee.txt' file and sort on age in descending order using Heap sort, Merge sort, Quick sort and write the sorted data to another file 'sortedemponage.txt'. If age is equal, using names in alphabetical order.

### EXERCISE 1-4 (ADVANTAGE – NOT REQUIRED).

Sort an arbitrary two-dimensional array in spiral order as following image:

1	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

## EXERCISE 1-5 (ADVANTAGE - NOT REQUIRED).

(Ref: <http://users.utcluj.ro/~jim/DSA/Resources/LabCode/DSALab.pdf>)

Performs the following operations encountered when processing data for an admission exam:

- a) Calculate the average scores for the candidates
- b) Assign places to the accepted candidates, based on  $m$  options they may have, and print the list.
- c) List all rejected candidates in descending order.

The exam is composed of two tests, graded with reals in the range  $[1, 10]$ . When average scores (truncated to two decimal positions right of the point) are equal, the score of the first test and then the scores of the second test is used to decide position. If equality persists, increase the number of available positions from a certain option.

### Input:

- number of options,  $m$  alone on one line
- pairs *option maximum number of admitted candidates*, separated by blanks, each pair on one line
- candidate data, one candidate on each line:

*Name,score<sub>1</sub>,score<sub>2</sub>,opt<sub>1</sub>,opt<sub>2</sub>,... ,opt<sub>m</sub>*

E.g.

### Input:

4

1 25

2 30

3 35

4 20

"Doe John",9.30,9.80,4,2,1,3

"Doe Jane",9.70,9.70,1,2,4,3

...

### Output:

Successful candidates for option 1

1. Doe Jane 9.70

...

...

Successful candidates for option 4

1. Doe John 9.55

Unsuccessful candidates

1. Jones Jim 4.99

...