# Advanced Programming Techniques

Aurel Vlaicu University Arad

## Practical Session #3

Applications of sorting.

**Exercise 1.** Read and understand the top interview questions regarding sorting found at `https://www.geeksforgeeks.org/top-sorting-interview-questions-and-problems/`.

**Exercise 2.** Assume we have $2n$ players that we need to partition into two teams of $n$ players. Each player has a number indicating how good he/she is in the game. Provide strategies that:

- Divide players in the most unfair way: the "talent" difference between the two teams is the largest possible.
- Divide players in the most fair way: the "talent difference between the two teams is the smallest possible.

Find the complexities in both cases and implement your solution.

**Exercise 3.** Given two sets $S_1, S_2$ of size $n$ and a numbe $x$ find a $O(n \log n)$ algorithm which finds whether there exists a number $x_1 \in S_1$ and a number $x_2 \in S_2$ such that $x_1 + x_2 = x$.

**Exercise 4.** Propose a reasonamble method for solving the following problems. Give the order of the worst case complexity for your methods.

1. You are given a pile of thousants of telephone bills and thousands of checks sent in to pay the bills. Find out who did not pay.
2. You are given a list containing the title, author, call number and publisher of all the books in a school library and another list of 30 publishers. Find out how many of the books in the library were published by each company.
3. You are given the records of the university library showing the statistics containing for each book the person which borrowed it. Determine how many distinct people checked out at least one book.

**Exercise 5.** Use the partitioning idea of quicksort to give an algorithm that finds the *median* element of an array of $n$ integers in expected $O(n)$ time.

In the following, the objective is to code, compare and analyze the following sorting algorithms:

1. InsertionSort
2. QuickSort
3. MergeSort
4. HeapSort
5. others

**Exercise 6.** (Experimental analysis) (a) Let $N$ be the size of the array to be sorted. Compute empirically the time the algorithm takes compared to $N$:

| $N$ | INSERTIONSORT | QUICKSORT | MERGESORT | HEAPSORT |
|---|---|---|---|---|
| 10 | | | | |
| 100 | | | | |
| 1000 | | | | |
| 10000 | | | | |
| 100000 | | | | |
| 1000000 | | | | |

For each $N$ run the sorting algorithm a fixed number $k$ of times and compute the average running time.