University of Science and Technology of Hanoi *** **Final Examination** **Subject: Algorithms and Data Structures** Sheet: 02 ICT + CS only No of pages: 02		***Intake***: *BI11* ***Academic year***: *2021–2022* ***Date***: *26/11/2021* ***Time***: *90 minutes* ***Important instructions*** *(according to lecturer's decision)* 1. Only the course slides and your own exercises' code are allowed in the examination venue. 2. Copy or using Internet will lead to heavy penalty	
Pathway coordinator		**Lecturer (or Head of Subject)**	**Dr. Đoàn Nhật Quang**
Student name		**Student's ID**	

Follow this instruction:
- Create a folder "ADS_YOURNAME_STUDENTID" in the Desktop.
*- Create the source files **question1.c** (or cpp) and **question2.c** for the corresponding problems.*
*- **Remove the executable files** (.exe) and **zip** all your source codes, submit to the Google classroom:* https://classroom.google.com/c/MzgyODQxMzI1Mzky?cjc=6khijwf
- Verify your name in the files and mails, un-named or incorrect-name files lead to 0.

Problem:
In this problem, we would like to re-implement a selection sort to sort an array of numbers with some improvements.

Selection Sort divides the input list into two parts: the sublist of elements already sorted and the unsorted sublist of elements remaining to be sorted.

- find the smallest element in the unsorted sublist
- swap this element with the leftmost unsorted element, it equivalents to move this element from the unsorted sublist to the sorted one,
- continue to proceed all elements in the unsorted sublist.

Question 1: (12 pts)

- Propose a recursive algorithm (pseudo-code) for the above **Selection Sort (**combined with iteration if necessary**)**. (3pts)
- Implement the proposed pseudo-code using C/C++ (7pts)
- Calculate the complexity of your program (Best scenario, Worst scenario, Average). Justify your answer. (2pts)

Question 2: (8 pts)
We would like to improve the Selection Sort by using **Binary Recusion**. The principles are as following:

Sorted part 1	Unsorted part	Sorted part II

- Find the minimum and maximum recursively from the unsorted part (two recursive functions).
- Put the minimum at the end of the sorted part I (which is empty at the beginning)

- Put the maximum at the front of the sorted part II (which is empty at the beginning)

- Implement the proposed pseudo-code using C/C++ (6pts)
- Calculate the complexity of this algorithm. Justify your answer. (2pts)

--END --