# 2023_DS_Fall_Homework 2

## Notice

**The deadline is 2023/12/21 23:59.** Homework should be submitted as a c source file, not an executable file. In your homework assignment, read the input from stdin and write your output to stdout. The file's name should be hw2_p1.c.

**Execution environment and Constraint.**

- CPU core: 1
- Memory: 2 GB
- Execution time limit: 1 second
- C Compiler: GCC
  - compiled with -O3 -std=c11 -Wall
- C Standard: C11
- Use header file only from C Standard Library
- OS: Linux 22.04.1 LTS

## Problem 1 : Priority Queue (3%)

Write a C function to create an empty F-heap and support the following instructions. Each line in the input file represents one instruction.

1. insert x val : insert an element with key x
2. extract : **print out the minimum** in the heap and delete it
3. delete x val : delete the node which has key x and value val
4. decrease x val y : decrease the key by y on the node which has key x and value y
5. quit : terminate the program

Note that all operations must leave behind properly structured F-heaps. Your functions for (3) and (4) must perform cascading cuts.

**Constraints**

- $-2147483648 \leq x$, val $\leq 2147483647$
- $1 \leq y \leq 2147483647$
- $2 \leq n \leq 10^5$, $n$ is number of instructions
- The key after decreasing will not exceed the boundary of 32-bit signed integer.

## Problem 2 : Efficient Binary Search Tree (10%)

Program the search, insert, and delete operations for red-black trees.

1. search x : Print out the ***color*** of the tree node if the element x exists. If the element x does not exist, print out Not found\n.
2. insert x : Add an integer x to the red-black tree. If x already exists, do nothing.
3. delete x : Delete the element x if the element x exists.

(Note: The textbook did not describe how to implement delete on red-black tree. Students can write your own delete **to satisfy the constraint** of the red-black tree.)

The instructions are insert, search, delete and quit. Instruction quit means you should terminate your program.