Data Structure and Programming, Fall 2018
Programming Assignment #5

by r07943102@ntu.edu.tw

Due 2 p.m., January 7, 2019

1 Problem Statement

Given a number K and a set of integer values, what is the minimum difference between the maximum and the minimum of all possible K-element subsets?

A k-element subset is a subset of a set on elements and contains exactly k elements. For example, all possible 3-element subsets of the given set $\{1, 13, 15, 24, 44\}$ are $\{1, 13, 15\}$, $\{1, 13, 24\}$, $\{1, 13, 44\}$, $\{1, 15, 24\}$, $\{1, 15, 44\}$, $\{1, 24, 44\}$, $\{13, 15, 24\}$, $\{13, 15, 44\}$, $\{13, 24, 44\}$, and $\{15, 24, 44\}$. The difference between maximum and minimum of the above subsets are 14, 23, 43, 23, 43, 43, 11, 31, 31, and 29, respectively. Therefore, the answer is 11.

2 Input/Output Specification

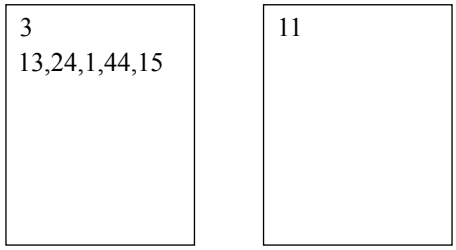

Figure 1: (a) An input example. (b) The output example of (a).

2.1 Input Format

The input consists a number K and an integer sequence separated by a newline. The number K in the first line is the size of the subset. The integer sequence in the second line is a set of integer values separated by a comma. Figure 1(a) shows an input example.

2.2 Output Format

The output contains only a number, the minimum difference between maximum and minimum of all possible K-element subsets. Figure 1(b) shows an output example.

3 Command-line Arguments

Please follow the command-line arguments as below.
Usage:

python3 programming_hw5.py [input file] [output file]

Example:

python3 programming_hw5.py input.txt output.txt

4 Evaluation

To check the correctness of your implementation, 10 public cases are provided for you. In the final evaluation, we will use the 10 public cases and 10 hidden cases (5% for each case) to evaluate your code. The sizes of hidden cases are similar to those of the public cases. The runtime limit for each case is 60 seconds.

5 Submission

- Please compress your programming_hw5.py (.zip or .tar) and upload it to CEIBA.
- Please submit your code before **2 p.m. on January 7, 2019**.

6 Requirement

- You may need to implement an efficient sorting algorithm (e.g., quick sort, merge sort, or heap sort).
- **Do not** use any python built-in sort function. You need to implement the algorithm by yourself.