Algorithms 2024

Homework #2

- The somewhat dumb Professor Kim, who teaches a tedious algorithm class, has been pondering each semester on how to assign grades optimally based on students' final scores, but he has not found a good solution yet. Thanks(?) to so annoying Chat-GPT, Professor Kim, who had no homework to assign, decides to *kill two birds with one stone* by making this problem into homework. (Darn it! This happens!)
- Let's help Professor Kim through this assignment. Whether Professor Kim will appreciate it is anyone's guess. So far, Professor Kim has been using manually identified intervals with significant score differences as grade distinction points.

Homework #2

- **(Method 1)** Let's first implement Professor Kim's method properly:
  - Take the scores of n students as input, sort them in descending order, and then divide them into k groups.
  - Assume that the lower group numbers correspond to higher scores.
  - Starting from $i = 1$ up to k-1, calculate the differences between the minimum score of the i^{th} group and the maximum score of the $(i+1)^{\text{th}}$ group, and maximize the sum of these differences.
  - However, each group must have at least one student.

Homework #2

- **(Method 2)** Now, let's implement a better method:
  - Similarly, take the scores of n students as input, sort them, and then divide them into k groups.
  - Again, assume that lower group numbers correspond to higher scores.
  - It would be ideal if the scores of students within each group were similar.
  - Therefore, calculate the variance of student scores in each group, and aim to minimize the sum of these variances across all groups.
  - However, each group must have at least one student, and the variance of a group with only one student is considered to be 0.

Input

- Take input from standard input.
  - The first line contains n and k.
  - The next line contains the scores of n students in ascending order of enrollment numbers.
  - Here, n is a number greater than or equal to k.
  - Each score can range from 0 to 1,000, and ties are possible.
  - Surprisingly, n can be as large as 10^4.
  - k is a positive integer less than or equal to 12. (Note: depending on the school, each grade (A, B, C, D) may be divided into 2 or 3 categories)

Output

- Print to standard output and write to files simultaneously.
  - Print the (maximum) sum of differences in Method 1 on the first line of standard output.
  - Write the groups from Method 1 to a file (filename: Partition1.txt). The first line of the file corresponds to Group 1, and the k^{th} line corresponds to Group k. Each line lists "student number (student score)" in ascending order of student numbers.
  - Print the (minimum) sum of variances in Method 2 on the second line of standard output (rounded to three decimal places).
  - Write the groups from Method 2 to a file (filename: Partition2.txt). The first line of the file corresponds to Group 1, and the k^{th} line corresponds to Group k. Each line lists "student number (student score)" in ascending order of student numbers.

Example

- Input (standard/console input)

```
15 3
50 50 10 20 50 10 50 50 20 20 50 50 50 50 10
```

- Output (standard/console output)

```
40
0
```

- Output (filename: Partition1.txt)

```
1(50) 2(50) 5(50) 7(50) 8(50) 11(50) 12(50) 13(50) 14(50)
4(20) 9(20) 10(20)
3(10) 6(10) 15(10)
```

- Output (filename: Partition2.txt)

```
1(50) 2(50) 5(50) 7(50) 8(50) 11(50) 12(50) 13(50) 14(50)
4(20) 9(20) 10(20)
3(10) 6(10) 15(10)
```

Example

- Input (standard/console input)

```
15 3
50 85 10 35 45 15 75 80 25 30 55 60 65 70 5
```

- Output (standard/console output)

```
20
197.917
```

- Output (filename: Partition1.txt)

```
1(50) 2(85) 5(45) 7(75) 8(80) 11(55) 12(60) 13(65) 14(70)
4(35) 9(25) 10(30)
3(10) 6(15) 15(5)
```

- Output (filename: Partition2.txt)

```
2(85) 7(75) 8(80) 13(65) 14(70)
1(50) 5(45) 11(55) 12(60)
3(10) 4(35) 6(15) 15(5) 9(25) 10(30)
```

Discussion

- If there are restrictions on the number of students in a group, mention in detail how your approach would change, in your report.
  - For example, the total number of students in Groups 1 and 2 cannot exceed 30% of n. Also, the total number of students in Groups 1, 2, 3, and 4 cannot exceed 70% of n.

Due Date

- Soft deadline: **May 13, 2024**
- Hard deadline: May 19, 2024
  - But, will be deducted 10% per one day from your original score

Submission date	Deduction rate
May 14	10%
May 15	20%
May 16	30%
May 17	40%
May 18	50%
May 19	60%
May 20	100%

Notice (cont'd)

- You should observe the format of input & output exactly.

- You should submit a compressed file (**HW2_your-ID.zip**) containing the following two files to the web-site (https://klas.kw.ac.kr).
  - **HW2_your-ID.hwp/.docx/.pdf** // report document
  - **HW2_your-ID.c/.cc/.cpp** (or **.java**) // source code

Notice (cont'd)

- Source code
  - It should be compiled in
    - **C/C++ Language: Visual Studio 2010** or **higher**, or **gcc/g++**
    - **Java Language: not restricted**
    - **You should note your environment in your report.**
  - Your name and student ID should be noted at the top of your source files in the form of comment.
- Report
  - Free format.
  - But, it must include several examples for testing your program and your own discussion.
  - It will be an important factor for getting a good score.
  - Mention your programming language together with compiler.