Design and Analysis of Algorithms **Programming Assignment #2**
CS3230 **6 April 2016**

CS3230: Assignment 2
Police Office Allocation

1 Objective

The purpose of this assignment is for you to practice the following skills which you are expected to master in this course:

- Algorithm design;
- Correctness analysis;
- Efficiency analysis;
- Implementation of algorithms.

2 Problem

Background

There is a straight highway with villages alongside the highway. The highway is represented as an integer axis, and the position of each village is identified with a single integer coordinate. ***There are no two villages in the same position***. The distance between two positions is the *absolute value* of the difference of their integer coordinates.

Police offices will be built in some, but not necessarily all of the villages. Obviously, a village and the police office in it have the same position. For building the police offices, their positions should be chosen so that the total sum of all distances between each village and its ***nearest*** police office is minimal.

Your Task

You are to write a program which, given the positions of the villages and the number of police offices, computes the least possible sum of all distances between each village and its nearest police office. You can write your program either in C/C++ or in Java. The task has been set up in the **CodeCrunch** programming environment, which we are using for the submission of your program. You can access CodeCrunch for either Java or C/C++ through the following two links:

`https://codecrunch.comp.nus.edu.sg/task_view.php?tid=2600`

`https://codecrunch.comp.nus.edu.sg/task_view.php?tid=2601`

Input

Your program needs to read the input from STDIN. Each test case contains two lines: the first line contains two integers: the number of *villages* $V \in [1, 300]$ and the number of *police stations* $P \in [1, 30], P \leq V$ separated by a blank space. The second line contains V integers *in increasing order*. These V integers are the positions of the villages. For each position X it holds that $1 \leq X \leq 10,000$.

Note: We will test your program with some additional input cases.

Output

Your program needs to generate a plain text output written to STDOUT. For each test case, output ***the sum of all distances between each village and its nearest police office***. The positions of the police offices is not required.

Sample Input

The sample of the input is the following:
10 5
1 2 3 6 7 9 11 22 44 50

Sample Output

Your program should print the result to STDOUT as follows:
9

Note that your program's output will be scored **automatically** on **CodeCrunch**. It is therefore very important that your output format is as described above. Otherwise your output may be mis-judged.

3 Deliverables

- A Java or C/C++ program with the name "Main.java" or "main.c" (or "main.cpp") submitted via **CodeCrunch**. Please make sure that your program reads its input from STDIN and writes its results to STDOUT.
- A short report (12pt font, at most 3 A4 pages – max. 2 pages are preferred), containing the following:
  - Your name and student number;
  - A description of your algorithm (you can use pseudo-code or plain English);
  - A correctness proof for your algorithm;
  - A time complexity analysis for your algorithm;
  - (Optional) Any other information that you feel is important about your algorithm and implementation.

  Your report should be in PDF format and named as U007008X_Ex2.pdf, where U007008X is to be replaced by your student number. Remember to write your name and student number at the beginning of your report.

4 Submission

- Submit your program, Main.java, main.c or main.cpp via CodeCrunch before **23:59** on Sunday **17 April 2016**. (We will know your program from your login information.)
- Submit your U080887X_Ex2.pdf report to the IVLE Workbin folder called **Ex2** before **23:59** on Sunday **17 April 2016**.
- Late submission: submit to the folder **Ex2Late** before **23:59** on **18 April 2016**, but 30% marks will be deducted for such a late submission. The same applies for the submission of your program.
- Late submission: submit to the folder **Ex2Latest** before **23:59** on **19 April 2016**, but 60% marks will be deducted for such a late submission. The same applies for the submission of your program.
- No submissions will be accepted after **23:59** of **19 April 2016**.

5 Grading Policy

This exercise is worth 14% of your overall grade, and is graded on a 100 point basis, with 50 pts for your program and another 50 pts for your report.

Grading of Program

The program will be automatically graded on the *CodeCrunch* server. Therefore, please make sure that your program is compilable under CodeCrunch. **Note that we do not have time to debug your program! No marks will be given for this part if your program does not compile and run on the CodeCrunch server.** In addition, make sure your program can accept the correct input (refer to the sample input).

Note that we will let your program run for a maximum amount of time during scoring. Some programs never finish and hence our automatic testing procedure will terminate such programs after what we think is sufficient time to compute the output. If you use a brute-force approach to solve this problem, then your program may not finish all the test cases in time (see Hint 1 below).

An input file will be used for grading. The file contains 5 test cases (*i.e.*, it has 11 lines), each of which is worth 10 pts.

Grading of Report

The 50 pts of this part are distributed as follows:

- Efficiency of your algorithm (20 pts), in terms of time and space.
- Correctness proof (15 pts).
- Complexity analysis (15 pts).

You are expected to analyze the efficiency of your algorithm based on the number of villages, denoted by V. The number of police offices, denoted by P, is regarded as a constant. Therefore, you should analyze your algorithm both in terms of time efficiency and space efficiency assuming P is fixed, but V is increasing.

Hint 1: Consider a dynamic programming solution for this problem. Start with the case where only one police station needs to be allocated. Note that you only need to output the *sum of all distances between each village and its nearest police office*. You do **not** need to output the position of each police office.

Hint 2: A good algorithm for maximum marks has $O(V^2)$ time complexity with a low multiplicative constant, *i.e.*, the faster your algorithm the more marks you receive.

The main grading criteria are efficiency, correctness, clarity and rigor.

Please note: Plagiarism will not be tolerated!