CSC2003 Assignment 4

CSC2003 Teaching Group

April 23, 2024

Important Notes

1. The assignment is an individual project, to be finished on one's own effort.
2. The work must be submitted before 6 p.m., May 12nd, 2024 (Sunday), Beijing Time. 20% mark deduction will be given for late submission within 2 days, and 0 for even later;
3. Plagiarism is strictly forbidden, regardless of the role in the process. Notably, ten consecutive lines of identical codes are treated as plagiarism. Using AI to directly generate code will also be regarded as plagiarism. Depending on the seriousness of the plagiarism, $30\% - 100\%$ marks will be deducted.

Marking Criterion

1. The full score of the assignment is 300 marks.
2. Three java programs are to be submitted. Each program will be evaluated with several unseen test cases. A submission obtains the full score if and only if both programs pass all test cases.

Running Environment

1. The submissions will be evaluated in the OJ system running Java SDK 21. It is the students' responsibility to make sure that his/her submissions are compatible with the OJ system.
2. The submission is only allowed to import four packages of (java.lang.*; java.util.*; java.math.*; java.io.*) included in Java SDK, and StdIn / StdOut from textbook. No other packages are allowed.
3. All students will have an opportunity to test their programs in the OJ platform prior to the official submission.

Submission Guidelines

1. You will get your grade only if you submit your code both on OJ and on bb on time.
2. For bb submission, you need to **directly** upload your java file on bb. That is, your submission should be *Vector.java*, and *StringSorter.java*. Wrong submission format will receive 10% mark deduction.
3. Inconsistency with or violation from the guideline leads to marks deduction.
4. All students are reminded to read this assignment document carefully and in detail. No argument will be accepted on issues that have been specified in this document.

Programs

There are 2 independent programs in this assignment, and each is worth 150 points.

Vector

Write a Java program (Vector.java) according to the following tutorial:

1. Create a class "Vector" which is used to store an n-dimensional vector and do relevant operations. Write a constructor for "Vector" that takes an int array as parameter.

2. Write a static method "plus(Vector v1, Vector v2)" that returns the sum of v1 and v2, which is also a Vector. Also, write a static method "subtract(Vector v1, Vector v2)" that returns v1 – v2.

3. Write a non-static method "multiply(int a)" that returns the result of multiplying a Vector by the integer a.

4. Write a non-static method "norm(int p)" that returns the p-norm of a Vector as a double. The definition of norm is given below:

   p-norm [edit]

   Main article: L^p space

   Let $p \geq 1$ be a real number. The p-norm (also called ℓ^p-norm) of vector $\mathbf{x} = (x_1, \ldots, x_n)$ is[11]

   $$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}.$$

   For $p = 1$, we get the taxicab norm, for $p = 2$ we get the Euclidean norm, and as p approaches ∞ the p-norm approaches the infinity norm or maximum norm:

   $$\|\mathbf{x}\|_\infty := \max_i |x_i|.$$

   The p-norm is related to the generalized mean or power mean.

   (https://en.wikipedia.org/wiki/Norm_(mathematics)#p-norm)

   Overload the method so that it will return the infinity-norm of the Vector when taking no parameter.

5. Write a static method "dot(Vector v1, Vector v2)" that returns the dot product of v1 and v2 as an integer.

6. Write a static method "angle(Vector v1, Vector v2)" that returns the angle between v1 and v2 as a double, provided that v1 and v2 are not 0 vector.

7. Override the non-static method "toString()" that returns a String in the form:

   $$(v_1, v_2, \ldots, v_n)$$

How to test your code: Go to the problem on OJ, and click "Open Scratchpad"; Copy your code to the top-right textbox, and copy the sample input to the bottom-left textbox; Click "run pretest" to generate output. Note that in this program, the class name must be "Vector". You can also test your code locally by yourself. Input and output handling is implemented in the OJ for this problem, so you can focus on implement the Vector class.

An example of input	Expected output
2 2 3 5 6 0 5 6 8 plus 0 1 subtract 1 0 multiply 1 -3 norm 1 1 norm 0 infinity dot 0 1 angle 0 1	(2, 8, 11, 14) (-2, 2, 1, 2) (0, -15, -18, -24) 19.000 6.000 93 0.258

StringSorter

Develop a Java program from the given template. Based on the template, write 3 classes that extend class "SortStrategy":

1. AlphabeticalSort: sorted by typical dictionary order. That is, 0-9-A-Z-a-z, given that there's no characters other than English letters and digits.
2. LengthOrderSort: sorted by the length of string, from shorter to longer. If some of the strings have the same length, keep the original order.
3. CharacterFrequencySort: sorted by the frequency of a given character, from lower to higher. Frequency is the number of occurrences of a character. If some of the strings have the same frequency, keep the original order.

Then, complete the constructor and the "sortContent" function in the "StringSorter" class. The processing of input has been given in the template, so don't modify other parts.

Input: The first line contains 2 integers t, n, representing the sorting strategy and the number of strings. Then, for the following n lines, each line contains a string made up of English letters and digits. If $t = 3$, there's an extra line that contains a single character, which is the frequency to be considered in sorting.

Output: The sorted version of the strings, one per line.

For all test cases, $t \in \{1, 2, 3\}$, $1 \leq n \leq 30$.

An example of console input	Expected console output
1 3 abc aBc 0bc	0bc aBc abc
An example of console input	Expected console output
2 3 abcd aBc Abcd	aBc abcd Abcd
An example of console input	Expected console output
3 3 aaaA aaAA aaaB a	aaAA aaaA aaaB