

Exercise 2 – Expression Puzzle¹

COMP 6651 – Winter 2018

Dr. Tiberiu Popa

Introduction

This exercise will help you acquire practical experience with dynamic programming. You are asked to implement a dynamic programming algorithm that solves optimally the expression puzzle problem. Given a set of digits $S = \{S_i\}_{i=1..k}$ and an integer N , the expression puzzle problem is finding a string consisting of characters from S (you can repeat a character as many times as you need) and the special symbols “+” and “*” (also you can repeat them as many times as you need) such that an arithmetic evaluation of the resulting expression yields the number N . For example, if $S = \{2, 9\}$, $N=229$ can be obtained by creating a string by concatenating the digits 2, 2 again followed by 9: “229”. $N= 11$ can be obtained by the string “2+9”, $N =49$ can be obtain using the string “9+2*9+22”, etc. An optimal solution is a solution that has minimal character length (i.e. any other solution string has more or equal number of characters). For example, for $N=22$, both “2*9+2+2” and “22” are valid puzzle solution, but only the latter is optimal. Note that for some puzzles many optimal solutions may exist, and some other puzzles may have no valid solution.

Specifications

The input is specified in a file whose name is the first argument of the program. The first line contains an integer M specifying how many datasets are in the file. The reminder of the file encodes the datasets. Each dataset is encoded in one line. It starts with an integer K that indicates how many elements are in the set S , followed by the actual digits in S (you can assume that the digits do not repeat). The last number in the line is the integer number N . Note that K , $\{S_i\}_{i=1..k}$ and N are separated by spaces.

Here is an example:

```
6
2 2 9 229
2 2 9 11
2 2 9 729
2 2 9 49
3 1 4 7 21
2 4 7 6
```

The output is a file called whose name is the second argument of the program. Each line encodes the results of each test case. If a solution exists, the algorithm should output the length of the optimal sequence of the puzzle. If none solution exists, the program should output the character “N”.

For example, the output corresponding to the input above is as follows:

¹ This exercise is a simplified version of a problem proposed to me by one of the students.

3
3
5
8
4
N

What is given

No code is given for this exercise.

Submission

You have to implement your program in plain C/C++ in a file called *main.cpp* that has no dependency other than the standard C/C++ libraries available in a standard Linux system such as the one in the graduate labs. The code should compile using the command *g++ main.cpp* in any machine in the graduate labs.

You need to submit the file *main.cpp* **ONLY** using the EAS system. If you submit anything else you will receive a grade of 0. Submit the exercise under “**Assignment 2**”. The due-date is **March 9th, 5pm**. The deadline is soft in that we accept late submissions at a penalty of 20% (of the total marks) for every 24-hour period. For instance, if you submit at 5:01pm on March 9th, you can receive maximum 80% of the maximum grade. If you submit at 5:01pm March 14th, you automatically receive a grade of 0.

This exercise is individual.

Originality and Plagiarism

Some of the problems proposed in the course are “classical” in the algorithm design literature and, therefore, solutions may be available on-line. Please note that you are expected to do this problem individually and you are expected to produce an original solution. We run plagiarism tests and if your submission is red-flagged you are expected to explain it in detail to the instructor. Failure to comply with this request or to adequately explain your own code may result in filing a plagiarism case with the Dean of the Faculty of Engineering.

Evaluation and Testing

The code is evaluated automatically, but we do check if you implemented the right algorithm and if the code is original. You will receive 0 if the code does not compile. We run 3 test cases, you receive 1/10 marks if your code compiles, but it does not return the correct results on any of the test cases, 4/10 if it compiles and runs correctly on one of the test cases, etc.

You will receive a code of 0 if you did not implement a proper dynamic programming solution.

You may be asked to explain your code to the teaching assistant or the instructor. This may lead to a decrease in your overall grade for this exercise.