Computer Science 220, S2 2022

Assignment 4 (traversal and optimisation)

See Canvas for due dates

This assignment requires you to submit programs in Python that you have written yourself to the automarker, `https://www.automarker.cs.auckland.ac.nz`. Your implementation must be from first principles and cannot use an existing library methods that might solve the problem (eg performs graph operations etc).

The automarker runs on a Linux box. Read the automarker help and FAQ for more details.

Please submit only Python source code (`.py` extensions only).

1. **Arithmetic trees** *30 marks*

   You are given an input file with multiple pairs of input lines. The first line of each pair is a tree given as a predecessor array. The second line is the value at the corresponding node. Values at leaf nodes (nodes with no children) are integers. At non-leaf nodes, the two possible values are $+$ or $*$.

   The tree represents an arithmetic expression where the value at a non-leaf node u is the sum of values at the children of u in the case of $+$, or the product of values at the children of u in the case of $*$.

   You need to calculate the value at each node and output the calculated value at the root. The tree is not constrained to be binary.

   Input format: Input consists of m pairs of lines of comma separated values, so $2m$ lines in total. The first line is each pair is a comma separated list of integers representing a tree in predecessor array format where -1 represents null.

   The second line in each pair is a comma separated list of integers and the symbols $+$ and $*$. The ith item on the list is the value or operator at the ith node in the tree.

   For example:

   ```
   -1,0,0,0,1,1
   +,*,2,3,0,7
   2,0,-1,0
   +,3,*,3
   ```

   Output format: For each pair of input lines, output a line containing the value calculated at the root of the tree.

   For the example input above, output would be:

   ```
   5
   6
   ```

2. **Optimisation** *30 Marks*

Consider a digraph G with non-negative arc weights. Your task is to find the length of the shortest route from a specified source, s, to a specified destination, t, with the added complication that you would also like to pass through node u if doing so does not increase the route length by more than 25%.

For each input digraph, the program should output one of:

(a) the length of the shortest route from s to t via u so long as this length is no greater than 1.25 times the shortest path from s to t; or,

(b) the length of the shortest path from s to t if the route through u is too long; or,

(c) -1 if t is not reachable from s.

Any pair, or all three, of the nodes s, t, and u can be identical. All arc weights will be integers.

Input format: A sequence of one or more digraphs is taken from the keyboard (System.in). Each graph is represented by a weighted adjacency list, with non-negative integer weights. The first line is an integer n indicating the order of the graph. The second line has three integers representing the nodes s, t, and u. This is followed by n white space separated lists of adjacencies and weights for nodes labeled 0 to n - 1. The lists are sorted by node. The input will be terminated by a line consisting of one zero (0). This line should not be processed. The sample input below shows two digraphs, the first has node set $\{0, 1, 2, 3\}$ and weighted arc set $\{((0, 1), 2), ((0, 3), 6), ((1, 2), 0), ((1, 3), 5), ((2, 0), 0)\}$ with s = 0, t = 3, and u = 1. The second has node set $\{0, 1, 2\}$ and weighted arc set $\{((0, 1), 1), ((0, 2), 5), ((2, 1), 2)\}$ with s = 0, t = 2, and u = 1.

```
4
0 3 1
1 2 3 6
2 0 3 5
0 0

3
0 2 1
1 1 2 5

1 2
0
```

Output format: For each input digraph, output an integer on a single line representing the correct route length (either from s to t, or from s to u to t) or -1 if no route from s to t exists.

For the input above the output would look like:

```
7
5
```

Marking

The maximum number of submissions for each problem is fixed at 12.

Each problem has three test cases associated with it worth one third of the marks for that problem. Some of the test cases will be large to test for efficiency. You get full marks if you pass all test cases.