Middle East Technical University
Department of Computer Engineering

CENG 111

Fall 2019

Take Home Exam 4

REGULATIONS

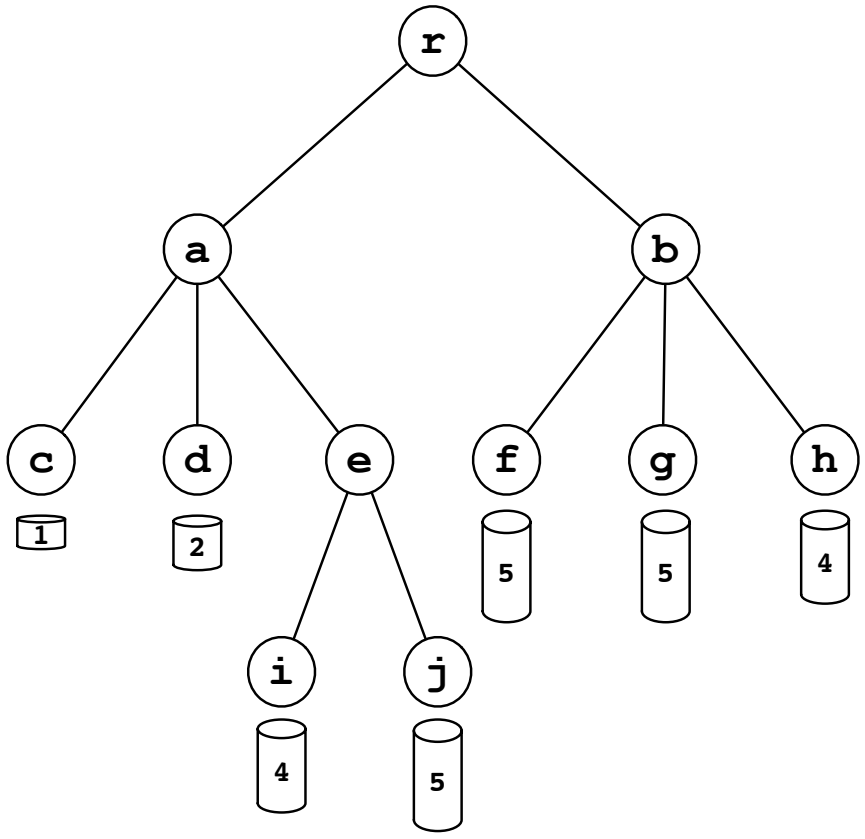
Due date: 16 January 2020, Thursday *(Not subject to postpone)*

Submission: Electronically. You will be submitting your program source code through a file which you will name as `the4.py` via `CengClass`. Resubmission is allowed (till the last moment of the due date), the last will replace the previous, provided you answer the interactive question positively.

Team: There is **no** teaming up. The take-home-exam has to be done/turned in individually.

Cheating: All parts involved (source(s) and receiver(s)) get zero. Furthermore, any cheating will result in disciplinary action with the claim of "cheating in examination".

PROBLEM



You are given a water pipe system organized into a tree. The edges are pipes and each vertex is a valve that controls whether (all) the children nodes will receive water or not. Initially all valves

are open (i.e.. water passes down to the children nodes). Any valve can be closed only once and cannot be reopened.

There are valves also located at the terminal nodes (leafs). We call them *taps*. Naturally, all specifications for the valves apply also for taps. The only difference is that there are no descendent nodes of a tap, but a cup is located under each tap. The size, which is a positive integer, of the cups vary.

The task is to exactly fill all cups (no partial full or empty cups nor spill over is allowed) by closing valves. The time flows in integer units and you are allowed to close as many valves as you desire between these time intervals. Closing valves do not take any time. Every open tap, fills its cup by one unit in each unit of time. This is absolutely independent of its siblings. Obviously, if a valve is closed, the water flow through all of its antecedents will cease (you do not have to close them as well).

You will be given the tree structure as a Python list (see next section). Here each valve is indicated by a string. You are supposed to produce a list of lists where each member indicates those valves to be closed at the end of the (passed) one unit time interval. If you decide that no valve is to be closed after that unit time interval you will include an empty list in the sequence. To get full credit for a problem instance you have to close minimal number of valves.

SPECIFICATION

- The Python representation that is used to denote the pipe system is as follows:

  for internal nodes: A list that starts with a string that labels the node. The following elements of the list are the representations of its children.

  for terminal nodes (taps): A list that starts with the string that labels the node which is followed by the size of the cup that is placed under that tap.

  For the configuration given in the figure above, the Python representation would be:

```
['r',['a',['c',1],['d',2],['e',['i',4],['j',5]]],
     ['b',['f',5],['g',5],['h',4]]]
```

- You are expected to write a function **`chalchiuhtlicue`** that will take one argument, the representation of a water pipe system and produce a list of lists. The element of the returned list has to correspond to a strict time order where the first element of the list is about the very past of the event (end of the first unit of time) and the last element is about the last unit of time, by the completion of which, all cups are full.

  For the example above, the call [the lines prefixed with >>>) and a valid return value is:

```
>>> chalchiuhtlicue(['r',['a',['c',1],['d',2],['e',['i',4],['j',5]]], \
['b',['f',5],['g',5],['h',4]]])
[[c],[d],[],[h,i],[r]]
```

- There is no restriction on the count if children of any node.
- There is at least one tap in the system.
- There might be more then one equivalent solutions you are free to chose any of them.
- No erroneous input will be used for evaluation.

- Make sure that your code can be called on inek machines with Python 2.7 Interpreter like below where three dots represent any valid argument for the function:

```
Python 2.7.15+ (default, Oct  7 2019, 17:39:04)
[GCC 7.4.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import the4
>>> the4.chalchiuhtlicue(...)
```

GRADING

- Comply with the specifications. Since your returned results will be evaluated automatically, non-compliant results will be considered as incorrect by our evaluation system.
- Suboptimal solutions which are solutions to the problem but close more valves than the minimum number will get partial point according to the following formula:

$$full\ point \times \frac{m}{c} \times \frac{m}{t} \tag{1}$$

  where m, c and t represent the minimum number of closing in optimal solution, c is the number of closed valves in your solution and t is the total number of valves for the given testcase.
- Your program will be tested with multiple data (a distinct run for each data).
- Any program that performs only 30% and below will enter a glass-box test (eye inspection by the grader TA). The TA will judge an overall **THE4** grade in the range of [0,30]. The glass-box test grade is not open to negotiation nor explanation.