WORD WIDE WEB - SOLUTION

WRITTEN BY DAVID HASHE AND IVAN WANG

ANSWER: PANCAKE STACK





We are presented with a word web with two kinds of edges (directed and undirected) and some nodes filled in. There is also a node labeled with a dot that has directed lines to all other nodes. This node, the title, and the given words all clue that the words in this word web are all top-level domains (TLDs). <u>Full list of TLDs here.</u>

SOLVING THE WORD WEB

The lengths of all unknown words are given through the number of blanks in the top section of the node. Sometimes the blanks are split over two lines for space reasons. This is not significant, but we did make sure to always split on a syllable boundary.

The directed edges represent concatenation; the directed edge "plus -> one" exists because "plus one"

is a common phrase. Not all possible concatenations are necessarily included as edges, because language is hard, but we tried to include the most obvious ones.

The undirected edges represent *country borders*. Country code top-level domains (ccTLDs) are an interesting subset of TLDs. There are some concatenation edges between ccTLDs and regular TLDs, but all ccTLD to ccTLD edges are undirected and represent that the two countries share a land border. All possible country borders edges are included. Some land borders are particularly surprising:

- France and Brazil share a border through the French overseas department of French Guiana.
- France and the Netherlands share a border through the Caribbean island of <u>Saint Martin</u>.
- Poland and Russia share a border through the Russian exclave of Kaliningrad Oblast.
- Azerbaijan and Turkey share a border through the Azerbaijani exclave of the <u>Nakhchivan</u> <u>Autonomous Republic</u>.

Filling out the word web reveals that the first letters of the words in the orange nodes, when read in reading order from left-to-right and top-to-bottom, spell out **FILLOMINO**.

SOLVING THE FILLOMINO

<u>Fillomino</u> is a logic puzzle usually played on a rectangular grid. A grid is just a graph with a regular adjacency structure, so we take the natural extension of Fillomino to an arbitrary graph. A few notes about our Fillomino variant:

- All edges are considered undirected for the purpose of connectivity. A -> B <- C could be a connected group of 3s.
- Some number groups have multiple numbers given at the start.
- Some number groups have no numbers given at the start.
- In particular, some 1 groups are unmarked.

This Fillomino puzzle has minimal clues: removing any one clue makes multiple solutions possible. Because of this, some advanced strategies are needed to solve the puzzle:

- Think about how many empty squares are available when deciding whether two numbers are part of the same group or different groups. For example, if you have five available spaces and two non-adjacent nodes marked 3, then they must be part of the same group of 3.
- If you have empty space and need to create one or more groups to fill the space, count the total number of available squares for an upper bound on group size and eliminate options based on the numbers of adjacent groups. For example, if you have a triangle of three available squares, a node marked 1 adjacent to one of the triangle nodes, and nodes marked 2 adjacent to the other two triangle nodes, then you know that all three triangle nodes must form a group of 3.
- For large groups, you can often mark some members based on the shape of the available space. For example, let's say you have a group of 5 that can expand in only two directions: _ _ _ _ _ _ _ _ _ . You can mark _ _ 5 5 _ _ because the group must expand at least one in each direction.

Extraction

There is a row of unconnected nodes below the word web. The combination of word length, Fillomino number, and (if present) text underneath identifies a specific node in the word web. Text is present underneath all of the ccTLDs because all ccTLDs have length two, which makes it hard to identify them uniquely with only a Fillomino number.

The most surprising text clue is ">1/2 speak a non-<u>Indo-European language</u>". While most people think of Paraguay as a Spanish-speaking country, it is officially bilingual, and the indigenous language Guaraní is <u>spoken by over 90% of the population</u>. Bolivia also comes close to meeting this criteria, but <u>only 43% of the population speak an indigenous language</u>.

Identifying the nodes gives us **SY RU PY FOOD PLUS CALL FR AM ES**. A pancake is a syrupy food, and the stack in a computer process is made up of call frames, so the answer is **PANCAKE STACK**.

AUTHOR'S NOTES

I've always been amused and impressed by the interesting range of gTLDs that exists, and the sheer number of concatenated pairs that work together led nicely to a puzzle idea. The idea of a Fillomino extract was proposed later as a neat variant on a logic puzzle, and felt like a cool way to mix together two different puzzle genres.

This puzzle bore some resemblance to the <u>Black Widow</u> web in Puzzle Potluck 3, which we solved after the initial ideation of the puzzle. While we wish we had time to implement a similar interactive answer-box (kudos to their web team!), we feel the style of answers (concatenation and borders as opposed to categories and semantic relations) were sufficient to differentiate the two.

On a technical note, the graph images were generated using a custom R+ggplot+ggraph script that consumed several CSV files specifying nodes, edges, and various styling parameters. This approach let us collaborate on the content over Google Sheets and then quickly render nice prototypes. Most of the concatenations were found by grep'ing through the entire English Wikipedia for two-word sequences that were both TLDs. Graph layout is a hard problem, and the Davidson-Harel algorithm from ggraph ("dh") that we used for prototyping tended to create graphs with lots of crosses, so for the final version of the puzzle we wrote a D3.js script that placed the nodes according to a force-graph but let us interactively move and fix nodes until we were satisfied with the output. That's how we generated the final layout that is used on the site.

We also implemented a solver for our extension of Fillomino using the Z3 theorem prover. This ended up being very challenging and may not have been the best approach, but it worked! It was important that we prove that our puzzle had a unique solution because Fillomino lets you do crazy things like insert new number groups and combine existing number groups, which means that a tiny change to one part of the puzzle can easily propagate throughout the puzzle. We also wanted to have a puzzle with minimal clues, but the main motivation was to make sure it was actually correct.