Newspapers – solution

Ivan Paljak

September 5, 2021

1 Branko survives if there is a cycle

The easiest observation in this problem is that the input graph at least has to be a tree in order for Ankica to have a successful strategy.

To prove this, we can generalize the explanation of the second example. More precisely, if we assume the graph contains a cycle, we can imagine Branko starting at any node of the cycle that is different from a_1. Since each node in the cycle has two neighbours that are also part of the cycle, Branko can choose a node in the i-th turn that's different from a_i.

2 A brute force approach

Notice that Ankica only needs to keep track of the set of nodes Branko can occupy before each turn. Of course, before the game, she assumes that Branko can occupy any node.

This leads us to represent the game as a graph where each node corresponds to a set of nodes that Branko can occupy at some turn. The graph will be directed, and the edge from node u to node v will mean that Ankica can make a guess such that if Branko could be located at a set of nodes represented by u before a turn, he could be located at a set of nodes represented by v after the turn.

Obviously, we want to find the shortest path from a node representing the set of all nodes to a node representing an empty set.

This approach was fast enough to solve the first subtask.

3 Branko doesn't survive on a chain

Now let's show that Ankica can always catch Branko on a chain.

Suppose node labeling is consistent with the second subtask, and let's color the nodes with even labels white, and nodes with odd labels black.

The cases when $N = 1$ and $N = 2$ can be trivially solved, we assume $N \geq 3$ in the rest of the section.

Ankica will make a total of $2N - 4$ guesses: $(2, 3, \ldots, N - 1, N - 1, N - 2, \ldots, 2)$. It is easy to show that she will catch Branko during the first $N - 2$ turns if he starts on a white node, and that she will catch him during the last $N - 2$ turns if he starts on a black node.

Suppose Branko starts on a white node. In that case, during the first $N - 2$ turns, Ankica and Branko will occupy a node of the same color. It's easy to see that in order for Branko to survive the i-th turn, it must be true that $b_i > a_i$. This is because $b_1 > a_1$ (otherwise he is caught in the first turn), and in order for b_i to be less than a_i, we would have to have some $b_j = a_j + 1$ ($j < i$), which is not possible because b_j and a_j must be of the same parity. Since $a_{N-2} = N - 1$, there is no node of the same color with greater label, and we can conclude that Branko will be caught during the first $N - 2$ turns if he starts on a white node.

If Branko starts on a black node, he will occupy a node of different color than Ankica during the first $N - 2$ turns. Since the $(N - 1)$-st turn repeats the node, Branko will occupy the same colored node during the last $N - 2$ turns. Thus, the last $N - 2$ turns when Branko starts on a black node are analogous to the first $N - 2$ turns when he starts on a white node, just in the opposite direction.

We will later show that this construction is indeed optimal.

A commonly submitted suboptimal solution for a chain with similar reasoning was $(1, 2, \ldots, N, N, \ldots, 1)$.

4 Branko survives on a star with three arms of length three (S')

Here we will show that Ankica cannot catch Branko on a specific star-shaped tree with 10 nodes. We will label the central node with label 10, the three arms with labels $(1, 2, 3)$, $(4, 5, 6)$, and $(7, 8, 9)$ as depicted below.

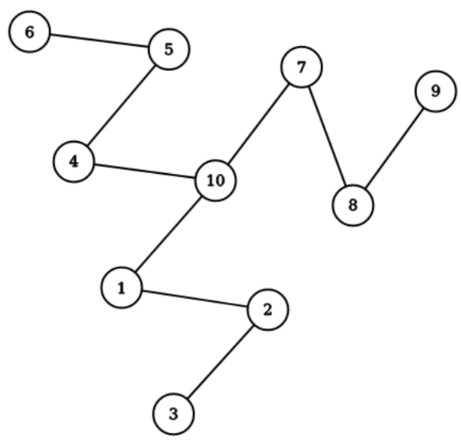

Figure 1: Star with three arms of length three

We will show that Branko can survive by spending his even turns either at the central node (10), or at the middle arm nodes $(2, 4, 8)$. Loosely speaking, Branko will attempt to stay as close to the central node as possible during the entire game.

Therefore, if Branko finds himself next to the central node, i.e. in one of nodes $(1, 4, 7)$, he will move to a central node in the next turn unless Ankica makes that guess. In that case, he will move to a middle arm node, i.e. one of $(2, 4, 8)$.

If Branko finds himself on a middle arm node during a turn, he will move towards node 10, i.e. to one of the nodes $(1, 4, 7)$, unless that is Ankica's next guess. In that case, he is forced on of the leafs $(3, 6, 9)$.

Obviously, if Branko finds himself at nodes $(3, 6, 9)$, he will be forced to move to nodes $(2, 4, 8)$ respectively.

The key idea is that if Branko finds himself in the central node, he will move to a branch which Ankica will not guess in the next turn, or the next branch in which Ankica wlil guess two closest nodes to the central in succession, i.e. $(1, 2)$, $(4, 5)$, or $(7, 8)$.

Notice that we only need to show that Branko will never be at a leaf node when Ankica guesses its neighbour on the next turn. In order for Branko to visit a leaf node, that means that Ankica must guess the node of the same arm next to the central one in that turn. If Ankica catches Branko on the next turn, that would imply her guessing two successive nodes from the previous section. For that to be possible, she would have to have guessed such consecutive nodes of some other arm after his last visit to node 10. This is in contradiction with the rules because during her guesses, he would have returned to the central node.

5 Branko doesn't survive on a tree that doesn't contain S'

In general, it should be easy to conclude that if Branko can't survive on a graph G, he also can't survive on a graph H with an induced subgraph isomorphic to G. In other words, Branko will escape in any tree containing S'.

Interestingly enough, it turns out that Ankica can always catch Branko on a tree that doesn't contain S'. We can show that by employing a similar strategy to the one explained in a section that deals with a chain.

Let's observe the longest path (diameter) of a tree, and assume it's nodes are labeled $1, 2, \ldots, l$, where l is the length of the diameter. We will also generalize the coloring from the chain analysis. A node will be colored white if his distance from node 1 is even, otherwise it will be colored black. Note that this coloring induces a bipartition of the tree.

Note that there must not be any node adjacent to nodes 1 and l, that are not 2 and $l-1$ respectively, as this would contradict with the diameter definition.

Note that there must not be any chain of length 2 with nodes not part of the diameter, that is connected to node 2 and $l-1$, as this would also contradict with the diameter definition.

Finally, note that there must not be any chain of length 3 with nodes not part of the diameter, that is connected to any node of the diameter because that would either contradict with the diameter definition or imply the existence of a subtree isomorphic with S'.

This means that all non-leaf nodes are either in the diameter or adjacent to a node in the diameter. This allows us to construct a strategy for Ankica that is very similar to the strategy employed for a chain. Here, Ankica will "move" from node 3 to node $l-2$ and back, but will also take into account all neighbouring non-leaf nodes.

More precisely, when after entering a node i on her route to node l, she will visit all non-leaf neighbours of i that are not $i+1$, before continuing to $i+1$. She will also visit the node i between each two successive visits to the neighbouring nodes. After visiting node l, she will reverse the walk back to node 2.

By similar reasoning to the one from section 3, we can inductively show that Ankica will catch Branko during the first half of the walk if he starts on a white node, and that she will catch him during the second half of the walk if he starts on a black node.

6 The optimal strategy

In the strategy from the previous section, we can differentiate between three types of nodes: leaves, inner nodes (i.e. nodes $3 \ldots l-2$), and nodes adjacent to inner nodes that are not leaves.

Alice never guesses the leaf nodes, 2 times guesses each node adjacent to the inner nodes, and $2f(v)-2$ times guesses a path node v with $f(v)$ neighbours that are not leaves. We will show that Branko can survive if any of these nodes is guessed any less than the described number of times.

Case 1. Node v adjacent to the inner nodes.
Since $a_i = v$ for at most one value of i, $a_i \neq v$ must hold for either all even or all odd values of i. Therefore, there is a way for Branko to move between v and a node adjacent to v the whole time without him being caught.

Case 2. Node v that is an inner node.
Branko will again try to stay close to v. Let u_1, u_2, $\ldots$, u_k be the neighbours of v, and u'_i be the neighbour of u_i that is not equal to v.

The number of times Ankica guesses v is at most $2k - 3$, therefore she guesses v on at most $k - 2$ even and at most $k - 2$ odd turns. We can assume she guesses v on at most $k - 2$ even turns. We will put Branko at v on every odd turn where $a_i \neq v$.

Now, we can divide Ankica's guesses into chains of longest sequences where she guessed v at every odd turn. In other words, we are looking at some a_i, a_{i+2}, $\ldots a_{i+2l}$ all equal to v, but $a_{i-2} \neq v$ and $a_{i+2} \neq v$. We can choose some u_j different from all $a_{i-1}, a_{i+1} \ldots, a_{i+2l+1}$. This is possible because $l < k - 2$. At all odd turns in this sequence, Branko will move to u'_j.

We now know all of Branko's whereabouts for all odd turns.

For even turns we consider where Branko is located at neighbouring odd turns. If on at least one of those turns he is at u'_j, he will be at u_j on the corresponding even turn. If at both neighbouring turns he is at v, he can simply choose any u_j that Ankica will not guess on a said even turn.

This shows that the strategy from the previous section is indeed optimal.

Disclaimer: The explanations in this editorial are meant to be illustrative, rather than rigorous. Rigorously proving each claim entails some cumbersome handling of special cases (e.g. small trees with $l \leq 2$), certain boundary checks (e.g. out of bounds indices), etc. We believe these shouldn't pose a serious problem to the reader that has grasped the main ideas.