EGOI 2025 Editorial - Currents

Task Author: Brian Lee Jun Siang

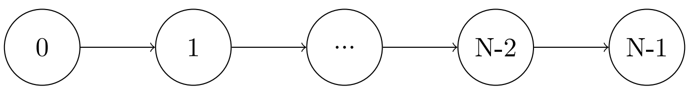
July 18, 2025

Introduction

In the task, a directed graph is given. Exactly once but at any point in time all edges are flipped. You shall calculate the shortest way from each node to the exit (node $N-1$ before the flip or node 0 after the flip), if the flip would be at the least advantageous time.

Subtask 1: $M = N - 1$ and $b_i = a_i + 1$

This subtask restricts the graph to be a path, like described by the graph below.



Looking at a node i. Before the graph is flipped, there is only one possible way to move through the graph, being to move closer to node $N-1$ node by node. After the graph is flipped, there again is only one possible way, being to step by step move closer to node 0.

The observation for this subtask is that the worst moment to get all edges flipped is one step before reaching node $N-1$. Therefore, a formula can be described for the shortest path to a exit when the flipping is in the worst moment: $dist[i] = N - i - 2 + N - 2$. The first part $N - i - 2$ describes the distance to the node $N-2$. The second part $N-2$ describes the way from node $N-2$ to the exit at node 0. This formula can be calculated and printed for each node using a simple loop.

Note that there is an edge case! If the graph only consists of two nodes, the shortest way is always 1 as in this case the worst case would be to go to node $N-1$ (node 1 in this case) and exit there.

Subtask 2: Each cave has a direct channel to cave $N-1$

Each node having a direct connection to node $N-1$ means that from each node the exit could be reached in one step. Therefore, the worst case for the shortest path is always when the edges get flipped before you even start moving.

For each node, the solution is the shortest way from this node to node 0 in the flipped graph. To calculate this, you can use the original graph and run a Dijkstra algorithm from node 0 resulting in the needed shortest paths.

Note that there again is an edge case! For node 0, the worst case would be to not flip the edges at all and having to reach node $N-1$ in 1 step. Flipping the edges immediately, like for all other nodes, would cause the distance to be 0.

Subtask 3: $N, M \leq 2000$

This subtask is aimed at any sort of solutions with a quadratic run time. Most importantly, this means an inefficient implementation of the full solution.

Subtask 4: Directed Acyclic Graph

The graph in this subtask does not contain any cycles. This allows us to use dynamic programming (DP). Notice that when you are at a node v that is not the exit, there are two options:

1. either the direction reversal happens now. In this case, we take the shortest path to cave 0.
2. The direction reversal does not happen now. Assuming we already know for each successor how long it takes in the worst case to get from there to an exit, we can now move to the successor which has the smallest such time.

The answer for node v is thus the maximum of the distance to 0 and the minimum of the results for the successors plus one. This leads to the following DP formula:

$$dp[v] = \max\left(\min_{w\ \in successors(v)}(dp[w]) + 1, dist[v]\right)$$

.

$dist[i]$ is the shortest path from node 0. This can be calculated beforehand using Dijkstra's algorithm.

The DP values can be computed in the inverse topological order. This ensures that the results for all successors are computed before the result of a node.

Full Solution

For the full solution, we can use the same insight as in the subtask before, but we no longer have a topological order. In order to still compute the results efficiently, we can adapt Dijkstra's algorithm to compute the results in order of increasing cost: we do not necessarily need to know the results for all successors of a node v for computing the result of v, it suffices if we know the result for the best possible successor!