Alternating Tree - Editorial

Difficulty:

Easy - Medium

Prerequisites:

Graph - ([Tutorial]())
Bitmask Dynamic Programming - ([Tutorial]())

Problem in Brief:

Given a graph with n nodes and m edges where each edge has a color(black or white) and a cost associated with it. Find the minimum spanning tree of the graph such that every path in the tree is made up of alternating colored edges.

Editorial:

First observation is that every such kind of spanning tree will be a chain. To prove it, suppose we have a tree that is not a chain and every path in it is made up of alternating edges. Then at least 1 node has a degree of 3. Out of these 3 edges, at least 2 will have the same color. The path using these 2 edges will not follow the conditions. This causes a contradiction. Hence, such kind of tree is always a chain.

Alternating Tree - Editorial

Now we want to find the chain with minimum cost that has alternating edges. This can be solved with bitmask dp :
dp[mask(2^n)][Node(n)][col_of_last_edge(2)]

where mask is the bitmask of nodes we've added to the chain, Node is the last node we added to the chain and col_of_last_edge is the color of edge use to connect Node. To transition from 1 state to another state we visit the adjacency list of last node and use those edges which have color != col_of_last_edge.

Time Complexity:

O(2^N * (M + N))

Similar Problems:

First

Second