Dead Ends

Akshat Kumar, Dhruv Yalamachi, Evan Zhang

October 29, 2019

1 Introduction

The game of Dead End is as follows. Players are given a directed graph. The first player chooses a starting vertex. The second player moves to any adjacent vertex (following the direction of the edges) and removes the edge they traversed to get to that vertex from the graph. The first player does the same, moving from the new location, and so on, alternating turns until some player reaches a dead end. The last player to successfully move wins. In this problem, you will analyze this game for various directed graphs.

For analyzing this problem, we assume that both Player 1 and Player 2 are playing optimally. Additionally, every graph in every section except for Section 7 is simple, or in other words, there is only one directed path from a vertex to another, with no multiple edges or loops.

Foundationally, we started by compiling a list of graphs we wanted to analyze, and then proceeded to analyze those graphs. In Section 8 of this paper, we proposed a slightly different problem and analyzed graphs there that would have been trivial in the original problem. We hope that this paper will set a good foundation for work done on this problem.

2 Non-Cycle Graphs

2.1 Player 1 wins if there is a vertex for which there is no directed path leaving it

Proof: If there is a vertex which there is no directed path leaving it, then Player 1 can pick that vertex to start with. This leaves Player 2 with no legal moves, and so Player 1 always wins in such a situation.

2.2 Player 1 wins in all directed non-cycle graphs

Proof: We prove this by contradiction. Start with two connected vertices, one of which is directed to the other. One of these vertices does not have a directed path leaving it, and so to ensure Player 1 doesn't win, we have to draw a directed path out of the original vertex without a directed path out of it (Figure 1).

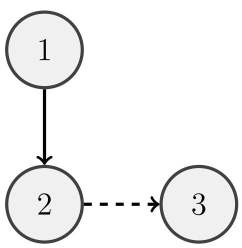


Figure 1: Continuous addition of directed paths

This then creates another vertex without a directed path leaving it, and so to ensure Player 1 still doesn't win, another directed path out of that vertex must be drawn. As long as there is no directed path that makes the graph a cycle, this pattern will continue on infinitely, and the resulting graph will always have at least one vertex that does not have a directed path exiting out of it. Hence we have arrived at a contradiction, as there is no way to ensure Player 1 doesn't win, and so Player 1 will always win.

3 Singular Cycle Regular Graphs

A singular cycle regular graph is any regular graph that has one and only one directed path that results in a cycle.

3.1 All k-length singular cycle regular graphs have k total edges

Proof: This is easy to see from an one-to-one correspondence of vertex to edge for singular cycle regular graphs.

3.2 If the graph is a singular cycle regular graph and has an odd number of edges, then Player 2 will always win

Proof: Because the graph is a directed cycle, it is arbitrary which vertex Player 1 starts at. Player 2 will proceed to get rid of the first edge, and then player 1 would get rid of the second edge, and so on. Hence, Player 2 would win if the graph had an odd number of edges, as it last removes an odd numbered edge.

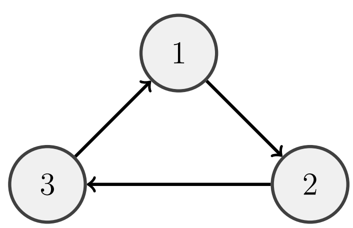


Figure 2: Example of a singular cycle regular graph with an odd number of edges

3.3 If the graph is a singular cycle graph and has an even number of edges, then Player 1 will always win

Proof: Refer to the the previous proof. Player 1 gets rid of the even numbered edges, and so they would win if the graph had an even number of edges.

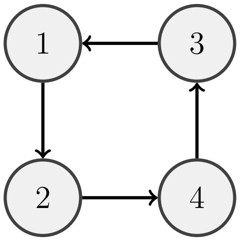


Figure 3: Example of a singular cycle regular graph with an even number of edges

3.4 Concluding Remarks About Singular Cycle Regular Graphs

Combining Theorem 3.1 with Theorems 3.2 and 3.3, we can conclude that for a k-length singular cycle regular graph, then:

$$\begin{cases} \text{Player 1 Wins, k mod } 2 \equiv 0 \\ \text{Player 2 Wins, k mod } 2 \equiv 1 \end{cases}$$

4 Double Cycle Graphs

A double cycle graph is a graph that consists of two cycle graphs that are connected via a common edge. Denote the receiving vertex of the intersection v and the exiting vertex of the intersection v*.

4.1 If both cycle sub-graphs have an even number of edges, then Player 1 will always win

Proof: Player 1 should choose one of the vertices that the two cycle sub-graphs share. The result is that Player 2 is forced to traverse along one of the even length cycle sub-graphs, which as seen in Theorem 3.3, means that Player 1 wins.

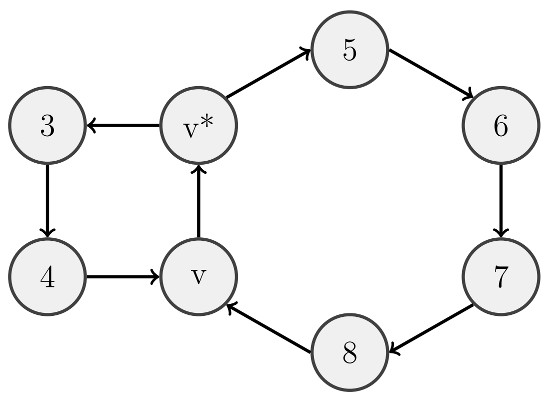


Figure 4: A double cycle with two even length cycle sub-graphs

4.2 If both cycle sub-graphs have an odd number of edges, then Player 2 will always win

Proof: No matter where Player 1 starts, one of the players will reach this vertex v*, so this is the pivotal point we analyze.

Case 1: It is Player 1's turn when the game reaches v*. Notice that iff there is an even number of edges remaining when the game reaches v*, then Player 2 wins. Otherwise, Player 1 can win by traversing along the odd length cycle sub-graph. Also notice that an odd number of edges must be deleted so that it is Player 1's turn at v*. Since the two sub-graphs share an edge, then both of them will always have an even number of edges remaining in this situation. Hence, Player 2 will always win in this case.

Case 2: It is Player 2's turn when the game reaches v*. In this case, Player 2 can continue along the sub-graph that the game started from, which would end in the player traversing along an odd-length cycle sub-graph, hence winning the game as seen in Theorem 3.2.

The only special case is if Player 1 starts at v*. In such a situation, the entire graph would be traversed; however, this doesn't change anything, as there is always an odd number of edges in the Double Cycle Graph with two odd-length components (due to the fact its odd plus odd minus one for the edge both graphs share).

Hence, we can conclude that Player 2 always wins if both cycle sub-graphs have an odd number of edges.

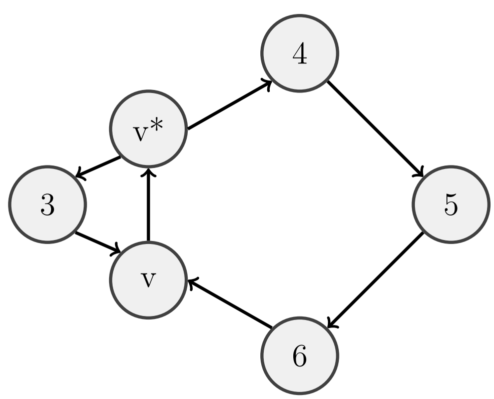


Figure 5: A double cycle with two odd length cycle sub-graphs

4.3 If one of the cycle sub-graphs has an odd number of edges and the other has an even number of edges, then Player 1 always win

Proof: If Player 1 starts from the vertex that is 1 vertex before vertex v, then Player 2 is forced to traverse along an even-length path, as they can either traverse along the sub-graph with an even number of edges or traverse the sub-graph with an odd number of edges with the parity shifted by one (or, in other words, a sub-graph that functionally has an even number of edges). Hence, Player 1 always wins in such a situation.

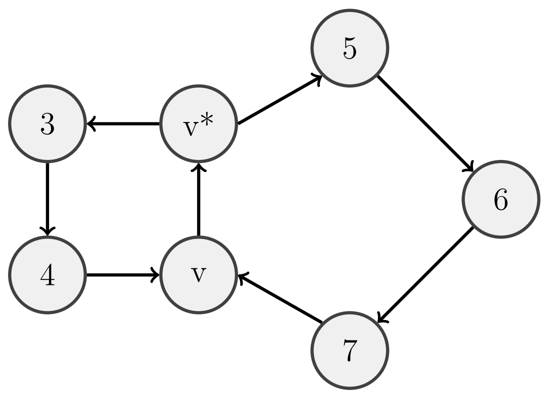


Figure 6: A double cycle one even and one odd cycle sub-graphs

4.4 For s-cycle graphs, Player 2 wins if all cycle sub-graphs are odd length

This has been proven for $s = 1$ and $s = 2$, as seen in Sections 3 and 4. More work and research would have to be done to prove this conjecture for all s-cycle graphs, as we purely conjecturize that such a hypothesis is true.

5 Singular Point k-cycle Graphs

Singular point k-cycle graphs are singular cycle graphs that are connected together by one point and consist of two cycle sub-graphs.

We analyze multiple cases for this type of graph: first, we start with the case of all cycles being of singular length k.

Within this case, we have 2 possibilities: k is even, or k is odd.

5.1 Conjecture: If both components of the Singular Point k-cycle Graph are of even length, Player 1 wins

Proof: Proceeding with the fact that both sub-graphs are of even length, we find that after every cycle, we return back to our starting point. For example if Player 1 were to start on the central meeting point, the game would continue and Player 1 will land back at the central point. This loop continues with cycle leading us back to our central point. If Player 1 were simply to start at this central point, we can guarantee they will win. This same analysis applies where ever Player 1 starts.

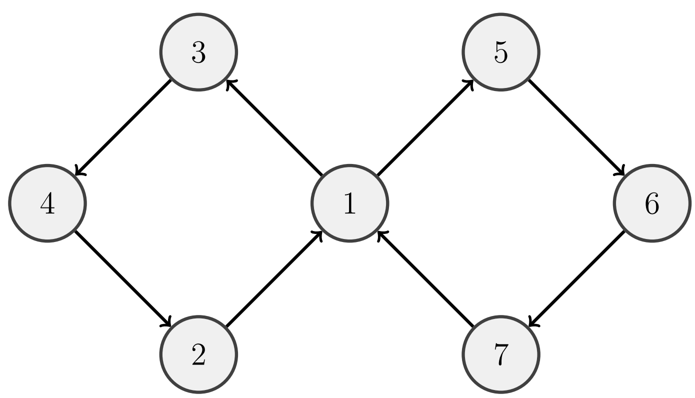


Figure 7: Singular Point k-cycle-Graph with two even length sub-graphs

5.2 If both components of the Singular Point k-cycle Graph are of odd length, Player 1 can always win

Proof: We then encounter our next scenario, that both sub-graphs are of odd length. Since every cycle is now odd, once we go through a cycle, we always end up with the OPPOSITE player being in control of the central point. Therefore, Player 1 is required to choose a point that will allow them to be the last player on the central point. If cycles are even, every cycle cancels itself; therefore, Player 1 can start in anywhere and still win in the end.

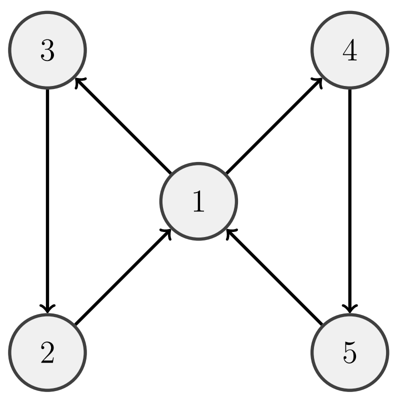
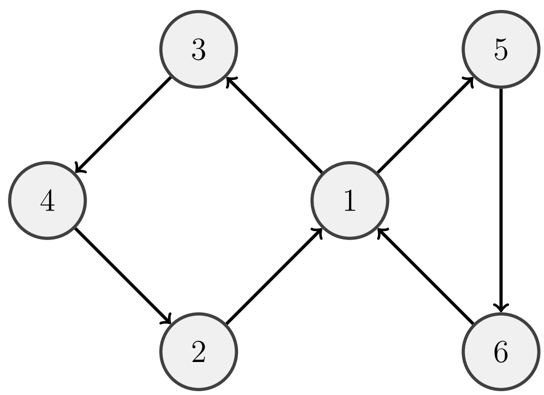


Figure 8: Singular Point k-cycle-Graph with two odd length sub-graphs

5.3 If one component of the Singular Point k-cycle Graph is of odd length and the other is of even length, then Player 1 wins

Proof: Similarly to before, Player 1 wants to control the center, because otherwise, Player 2 can choose one of the remaining sub-graphs to traverse and win. The key strategy here is that Player 1 should choose the vertex on the even length sub-graph that is right before the shared vertex.



5.4 If there are an even or odd number of k-cycles, where k is even but unique, Player 1 will always win

Proof: This relates back to our case of 2 even cycle graphs, every graph will return the central player back to their spot. In this way, every player who holds the central point at the start will automatically win, therefore, Player 1 must any point and will win.

5.5 If there are an even number of k-cycles, where k is odd but unique, Player 1 will always win

Proof: This situation relates back to our odd-odd cycle graphs. As proved in previous conjectures, odd graphs always result in the central player being replaced by the other after every cycle. Therefore, every odd graphs cancels each other in pairs. Since there are an even number of odd cycles, they all cancel out and are equivalent to a even-even graph to Player 1 will always win.

5.6 If there are an odd of k-cycles, where k is odd but unique, Player 2 will always win

Proof: This situation relates back to our odd-odd cycle graphs. As proved in previous conjectures, odd graphs always result in the central player being replaced by the other after every cycle. Therefore, every odd graphs cancels each other in pairs. Since there are an odd number of odd cycles, all but one cancel out and are equivalent to Player 1 owning the last center. However, one more cycle exists so Player 2 takes ownership of the center and therefore wins the game.

5.7 If there are an even of k-cycles, where k is odd or even but unique, Player 1 will always win

Proof: This situation relates back to our odd-even cycle graphs. As proved in previous conjectures, odd-even graphs always result in Player 2 winning at the end. This is due to the fact that the even cycles do not have an impact on central possession and therefore only the odd cycles impact the game. Since, in this case, there is an even number of odd cycles, we see this situation as that similar to an odd-odd graph with even, therefore Player 1 will always win.

5.8 If there is an odd number of k-cycles, where k is odd or even but unique, Player 2 will always win

Proof: This situation also relates back to our odd-even cycle graphs. As proved in previous conjectures, odd-even graphs always result in Player 2 winning at the end. This is due to the fact that the even cycles do not have an impact on central possession and therefore only the odd cycles impact the game. Since, in this case, there is an odd number of odd cycles, we see this situation as that similar to an odd-odd graph with an odd number of cycles. In that case, each pair of odd cycles cancel each other out and leaves one more odd cycles to make Player 2 win; therefore, Player 2 will always win.

6 Star Graphs

6.1 Given a graph is a star graph, Player 1 will always win

Proof: A trivial solution we found in our previous conjectures is that Player 1 will always win if there is a node that does not have any edges pointing out. We can see this is evident in a star graph. We define a star graph as one where every node has degree 1 except for the central node. In order to satisfy the condition that every node points out, we make every "outer" node have a degree 1 pointing into the central node. However, this makes the central node not have any edges that point out, so Player 1 chooses the central node.

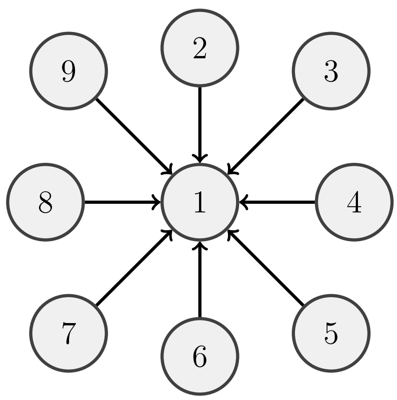


Figure 9: Star Graph with all nodes pointed centrally

On the other hand, if there is an edge pointing out, that means that Player 1 can choose an outer node and win.

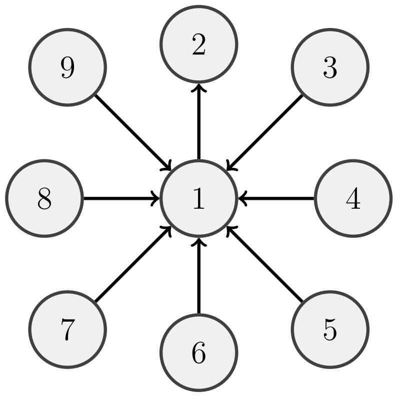


Figure 10: Star Graph with one node pointed outwards

7 Non-Simple Graphs with Two Edges between each vertex

To draw such a graph, we want to start with a graph we would normally have. We then want to draw another edge in to where the path is directed in both directions.

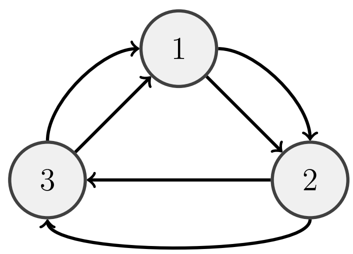


Figure 11: Example of a graph with two edges between each vertex

7.1 There is an even number of edges in such a graph

Proof: Since we draw an additional edge for every edge drawn before, we functionally double the number of edges, and so there is always an even number of edges in the graph.

7.2 It is impossible for there to be a hanging directed edge

Proof: The players can continue to play without ever turning back until they have exhausted all of the edges in one direction, in which they would have to turn back and finish the entire game. The players can also turn back early, but if they traveled along one directed edge in the path, then when they turn back they would have to travel along the other directed edge. Hence, it is impossible for there to be a hanging directed edge, which can be defined as a pair of directed edges where only one of the edges was traveled across.

7.3 Player 1 always win wherever they start

Proof: It is key to note that Player 1 will win if the game ends and the players have traversed an even number of edges, as seen in Theorem 3.3. Combining this with Theorem 7.1, we see that if the player traverses every edge on the graph, then Player 1 will win.

Now refer to Theorem 7.2. Since there cannot be a hanging directed edge, then the total number of edges traversed will always be even. Combining this with Theorem 7.1, this means that Player 1 will always win no matter where they start.

8 New Proposed Problem

Change: Player 1 is not allowed to start at a vertex which has no directed path exiting out of it. In other words, it is viable to analyze non-cycle graphs because Player 1 cannot trivially win by picking the un-exitable vertex.

8.1 Trees, Top to Bottom

Conjecture: Player 1 will always win.

Proof: Since Player 1 cannot start at an un-exitable vertex, then Player 1's goal this case is to force only an even number of edges being traversed.

Start with a vertex that is not directed to any other vertex (or, in other words, is at the bottom). Go backwards two edges and find that vertex, call it v*. If v* has a path that is directed to an un-exitable vertex, then Player 2 would win if Player 1 started from v*. If v* does not have a path that is directed to an un-exitable vertex, then Player 1 should pick v* to start from, as they would always win. For example, in the following graph, start at vertex 7. Go backwards two edges, and then call that vertex v*. Since v* does not directly connect to an un-exitable vertex, then Player 1 would win if they choose to start from v*.

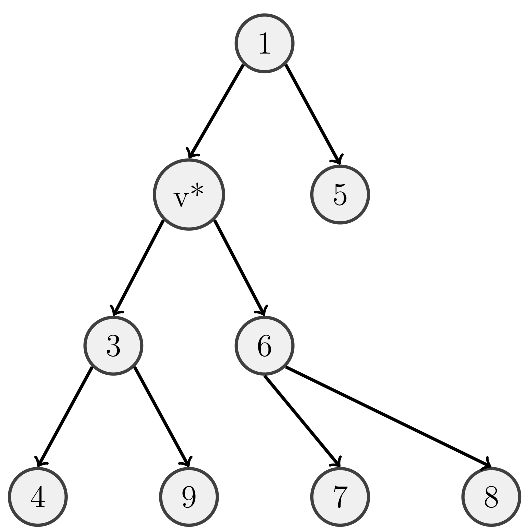


Figure 12: Example graph of a Top-Bottom tree

Hence the only way Player 2 can win is if each vertex that could possibly be v* (so they are 2 edges away from an un-exitable vertex) is directly connected to an un-exitable vertex (that is, there is one and only one edge in between them). Otherwise, Player 1 will always win.

For example, in the following graph, start from vertex 3, and then go backwards two edges and call that vertex v*. Since it is directly connected to an un-exitable vertex, then Player 2 would win if v* was chosen. Every other v* in this graph results in a similar situation, and so Player 2 would always win for this graph.

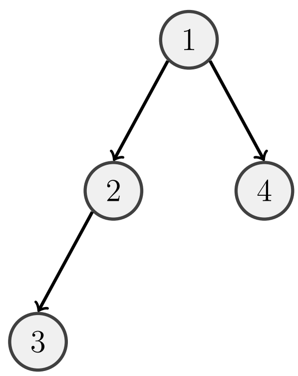


Figure 13: Example graph of when Player 2 would always win

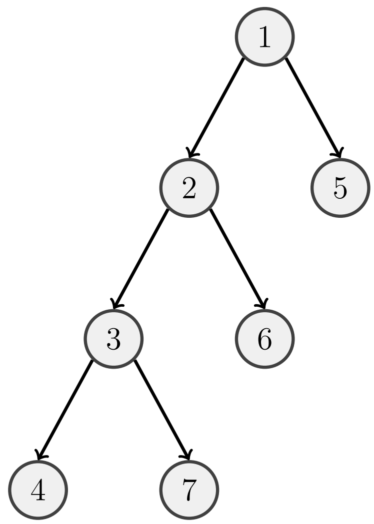


Figure 14: Example graph of a ”ladder graph” where Player 2 would always win

8.2 Trees, Bottom to Top

Conjecture: Player 1 will always win.

Proof: Here, Player 1 can simply start from the top vertex (or the vertex where all other paths converge at), go back 2 edges, and pick that vertex to start from. There is no option but for Player 2 to get rid of one edge, and then Player 1 can get rid of the final edge to win. For example, in the following graph, we can start from the top vertex, or vertex 1, and then go two edges backwards to pick a vertex, call it v**. If Player 1 chooses to start from v**, then they will automatically win, which is easily seen.

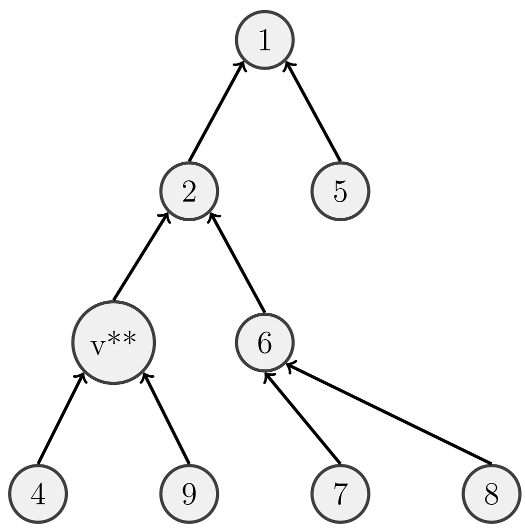


Figure 15: Example graph of a Bottom-Top tree

9 Concluding Remarks

We conclude this paper by remarking about the work that could be done to continue this game to be fully researched. One of the main things that could be done would be to analyze a k-number of cycle graphs, where instead of having one or two sub-components, we would have 3, 4, or something more. Then we could try and generalize this into all graphs that have such a shape. Furthermore, more work can be done on analyzing pseudo-random graphs that aren't trivial, or in other words, random graphs that still consisted entirely of cycles. Finally, graphs that were analyzed in this paper could be combined on a large scale and then analyzed. We hope our paper set a good foundation for work done on this problem, and hope to see great progress made on this problem in the future.