

Bidirectional Search Algorithm in Artificial Intelligence with Real Life Examples

Bidirectional search technique is an extension to breadth first search and depth first search. And it is a special kind of technique. Why is it special? 2 simultaneous search from initial to goal and backward from goal to initial. And stopping when 2 meets. We are searching from both the directions. Time complexity is the first important point over here. Time complexity over here as compared to BFS and DFS reduces. Time complexity is reduced as compared to DFS and BFS. Space complexity is $O(b^d)$ where b is a branching factor. Branching factor means how many nodes are there ahead and d means depth. Till how much depth you are going. And if we talk about space complexity also, then obviously how many number of nodes are being generated. That much only space complexity will be there. Depth first search generally works on one side. If I travel from here, I will go to this, then I went on this, then I went. Then if I don't get any path ahead, then I will do the backtracking. So obviously if you are starting from here, let's say you started searching from here, and this is your goal node. So you are searching from, in this direction. If you are using breadth first search, always then it is complete. Then there is the case of may or may not. Over here there is a simple example.

Time complexity as compared to BFS and DFS. It will be less, but if you are using depth first search, then in that case there is possibility of possibility of may or may not. But this gives guaranteed solution over here. So this is all about bidirectional search technique.