1. An algorithm with closed loop may cause tree search to go in a never ending cycle. It needs to know where it already visited.
2. *Line 3: Initialize the Explored set to be empty*

*Line 8: Add the node the explored set*

*Line 10: Only if not in the frontier or explored set*

1. Slide 23. The four components of a node in a search tree are: 1. **States:** Location of each of the eight tiles and the blank in one of the nite squares. N! 2. **Parent:** The node in the search tree that generated this node. 3. **Actions:** The action that was applied to the parent to generate the node. Movement of the blank space Up, Down, Left, or Right. 4. **Path Cost:** Given a state and action, it results in the resulting state. The path cost
2. The four ways to evaluate a search algorithm are: 1. **Completeness**: Is the algorithm guaranteed to find a solution when there is one? 2. **Optimality**: Does the strategy find the optimal solution? 3. **Time complexity**: How long does it take to find a solution? 4. **Space complexity**: How much memory is needed to perform the search?
3. Space and time complexity of BFS: The space complexity of BFS is dominated by the size of the frontier. Every state has B successors then the total nodes generated at each level (depth), N, is BN. There will be BN-1 in the explored set and BN in the frontier. So the space complexity is O(BN). Space complexity (memory) is a bigger problem than execution time, however execution time is still a major factor. If the problem trying to be solved has
4. Breadth First Search (BFS) does NOT account for the weight of the next node. BFS is UNinformed search. Also BFS “blindly” explores solutions so could waste time/space searching a path that will not yield a goal state. BFS could waste a lot of time exploring a path with a very high path costs that will not yield a goal state.
5. g(n) is **path cost** **function** which assigns a numeric cost to each path. Solution quality is measured by **path cost function** g(n). An optimal solution has the lowest **path cost** g(n) among all solutions. For the 8-puzzle problem the path cost, or g(n), is the same as the amount of steps in the path because each step costs 1.
6. BFS, DFS, and uniform cost search all use a tree data structure called a **search tree** with the initial state as the root, branches are the actions, and nodes correspond to states in the state space of the problem. They all use a data structure called the **frontier** which is a set of all leaf nodes available for expansion at any given point. Also, they all use a data structure called an **explored set** which remembers every expanded node. Also, they uses the data structure **queue** to store the frontier and explored set.
7. Uniform Cost Search will select the best path based on the frontier list determined by the graph. Greedy Best First Search will use a heuristic function to find the next best node and best path. Greedy Best First Search uses additional knowledge of the problem imparted to the search algorithm. **The implementation of GreedyBestFirst is identical to UniformCostSearch except the use of f(x) instead of g(x) to order the priority queue.** And f(x) is a heuristic function while g(x) is a path cost function.
8. To solve the puzzle, we need a heuristic function that never over estimates the number of steps to the goal. Two commonly used heuristics for the 8-puzzle problems are (H1) the number of misplaced tiles, and (H2) the sum of their distances of the tiles from their goal position. (H1) for a puzzle with all 8 tiles out of position would be equal to 8, and this would tell the algorithm that it needs at least 8 moves to get to goal state. (H2) which is sometimes referred to as the Manhattan distance is the sum of the horizontal and vertical distances that a tile is from its goal state. (H2) tells the algorithm how many steps it will need to get the goal state. For the start state below, (H2) is equal to 3+1+2+2+2+3+3+2=18. These two heuristics would imply that the algorithm would need to make (H1)+(H2) moves at least to get to the goal state, and the true solution cost is 26 steps so it is a solid estimate.

