N^2-1 puzzle problem

- The 8 puzzle is a well known sliding puzzle which was introduced by Sam Loyd in 1878.
- It uses a 3*3 board.
- This problem consists of 8 numbered tiles on a square frame with a capacity of 9 tiles in random order with one tile missing.
- These tiles are placed on a 3*3 chessboard
- The objective of this problem is to transform the arrangement of tiles from initial arrangement of the tiles in to the goal arrangement through a series of legal movements.
- The blank tile can move up, down, left and right and a tile can be moved only if there is an empty space next to it.

- There is always an empty slot in the initial arrangement.
- Legal moves are the moves in which the tiles adjacent to empty slot are moved to either left, right, up, down
- Each moves create a new arrangement, these arrangements are called as states of the puzzle
- The initial arrangement is called as initial state and goal arrangement is called as goal state.
- The state space tree for 8 puzzle problem is very large because there can be 9! different arrangements. A partial state space tree can be shown
- In the state space tree, the nodes are numbered as per the level.
- Each next move is generated based on empty slot positions.
- Edges are labelled according to the directions in which empty slot moves.

- The root node becomes the E-node
- We can decide which node to become an e-node based on estimation formula
- $\hat{c}(x)=f(x)+\hat{g}(x)$
- Where $f(x)$ is the length of the path from the root node to node ‘x’
- $\hat{g}(x)$ is the number of non-blank tiles which are not in their goal position for node ‘x’.
- $\hat{c}(x)$ is the lower bound on the value of $c(x)$

The 8-puzzle

- If the board size is 2*3 , then it is called the 5-puzzle problem.
- If the size of the board is 3*3, then it is called the 8 puzzle problem.
- If the size of the board is 4*4, then it is called the 15 puzzle problem
- 35 puzzle problem involves 6*6 chess board
- The object of the puzzle is to place the tiles in order by making sliding moves that use the empty space.
- The family of puzzle is called the N^2-1 puzzle, where N^2 corresponding to the no of block on the board

The 8-puzzle

- 8 numbered tiles on a square frame with a capacity for 9 tiles
  - – Given an initial arrangement, transform it to the goal arrangement through a series of legal moves

1	2	3
4		6
7	5	8

Initial Arrangement

1	2	3
4	5	6
7	8	

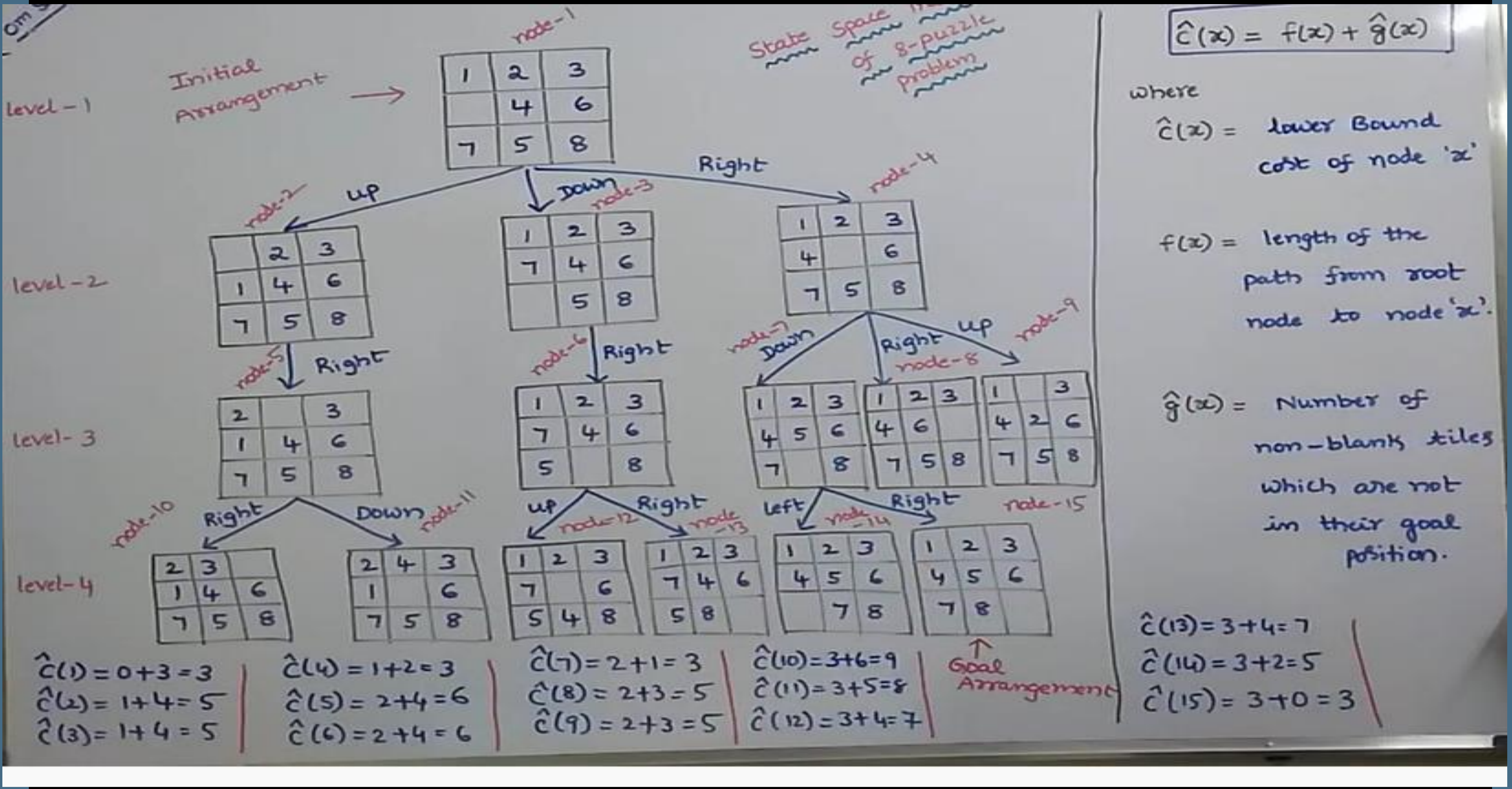
Goal Arrangement

The 8-puzzle

- Legal move involves moving a tile adjacent to the empty spot
    - – Four possible moves in the initial state above
    - – Each move creates a new arrangement of tiles, called state of the puzzle
    - – Initial and goal states
    - – A state is reachable from the initial state iff there is a sequence of legal moves from initial state to this state
    - – The state space of an initial state is all the states that can be reached from initial state
    - – Search the state space for the goal state and use the path from initial state to goal state as the answer
- Number of possible arrangements for tiles: 9!

The 8-puzzle

- In a state space tree nodes are numbered as per levels
- Each next move is generated based on empty slot positions
- Edges are labelled according to the direction in which empty slot moves.
- The root node becomes the E-node
- We can decide which node to become an E-node based on Estimation formula.


State Space Tree of 8-puzzle problem
Initial Arrangement
level-1
level-2
level-3
level-4
node-1
1 2 3
4 6
7 5 8
Up
Down
Right
node-2
2 3
1 4 6
7 5 8
node-3
1 2 3
7 4 6
5 8
node-4
1 2 3
4 6
7 5 8
node-5
Right
2 3
1 4 6
7 5 8
node-6
Right
1 2 3
7 4 6
5 8
node-7
Down
1 2 3
4 5 6
7 8
node-8
Right
1 2 3
4 6
7 5 8
node-9
Up
1 3
4 2 6
7 5 8
node-10
Right
2 3
1 4 6
7 5 8
node-11
Down
2 4 3
1 6
7 5 8
node-12
Up
1 2 3
7 6
5 4 8
node-13
Right
1 2 3
7 4 6
5 8
node-14
Left
1 2 3
4 5 6
7 8
node-15
Right
1 2 3
4 5 6
7 8
Goal Arrangement
ĉ(1) = 0+3 = 3
ĉ(2) = 1+4 = 5
ĉ(3) = 1+4 = 5
ĉ(4) = 1+2 = 3
ĉ(5) = 2+4 = 6
ĉ(6) = 2+4 = 6
ĉ(7) = 2+1 = 3
ĉ(8) = 2+3 = 5
ĉ(9) = 2+3 = 5
ĉ(10) = 3+6 = 9
ĉ(11) = 3+5 = 8
ĉ(12) = 3+4 = 7
ĉ(13) = 3+4 = 7
ĉ(14) = 3+2 = 5
ĉ(15) = 3+0 = 3
ĉ(x) = f(x) + ĝ(x)
where
ĉ(x) = lower Bound cost of node 'x'
f(x) = length of the path from root node to node 'x'.
ĝ(x) = Number of non-blank tiles which are not in their goal position.

The 8-puzzle

The 8-puzzle

Applications of Branch and Bound

- Branch and bound algorithms can be applied to optimization problems.
- We only deal with minimization problems, because a maximization problem is easily converted into a minimization problem by changing the sign of the objective function
- Searching for an optimal solution is equivalent to searching for a least cost answer node.
- The $\hat{c}(.)$ function will estimate the objective function value and not the cost of reaching an answer node.
- Any node representing a feasible solution will be an answer node.