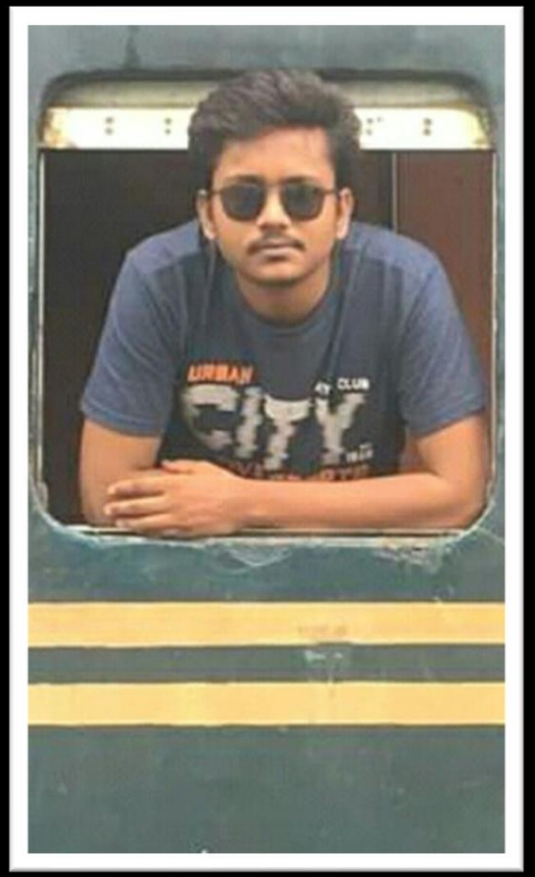
15 Puzzle

Teammates



Game Description

Initial state

2	3	4	0
1	5	7	8
9	6	10	12
13	14	11	15

Goal State

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Move

4 type of move:

Initial State :

2	3	4	8
1	5	0	7
9	6	10	12
13	14	11	15

UP

2	3	0	8
1	5	4	7
9	6	10	12
13	14	11	15

Right

2	3	4	8
1	5	7	0
9	6	10	12
13	14	11	15

Down

2	3	4	8
1	5	10	7
9	6	0	12
13	14	11	15

Left

2	3	4	8
1	0	5	7
9	6	10	12
13	14	11	15

Algorithm

IDA*

Iterative deepening A* (IDA*) is a graph traversal and path search algorithm that can find the shortest path between a designated start node and any member of a set of goal nodes in a weighted graph. It is a variant of iterative deepening depth-first search that borrows the idea to use a heuristic function to evaluate the remaining cost to get to the goal from the A* search algorithm.

IDA*

path	current search path (acts like a stack)
node	current node (last node in current path)
g	the cost to reach current node
f	estimated cost of the cheapest path (root..node..goal)
h(node)	estimated cost of the cheapest path (node..goal)
cost(node, succ)	step cost function
is_goal(node)	goal test
successors(node)	node expanding function, expand nodes ordered by g + h(node)
ida_star(root)	return either NOT_FOUND or a pair with the best path and its cost

IDA*

```
procedure ida_star(root)
    bound := h(root)
    path := [root]
    loop
        t := search(path, 0, bound)
        if t = FOUND then return (path, bound)
        if t = ∞ then return NOT_FOUND
        bound := t
    end loop
end procedure
```

IDA*

```
function search(path, g, bound)
    node := path.last
    f := g + h(node)
    if f > bound then return f
    if is_goal(node) then return FOUND
    min := ∞
    for succ in successors(node) do
        if succ not in path then
            path.push(succ)
            t := search(path, g + cost(node, succ),
bound)
            if t = FOUND then return FOUND
            if t < min then min := t
            path.pop()
        end if
    end for
    return min
end function
```

CODE

THANKS..!