Page No. | Date

Assignment - B4

Title :- N-Queens problem

Problem statement :- Implement n-queens problems (branch & bound)

Objective :-
1) Students will learn to implement n-queens problem.
2) Understand backtracking algorithm.

Outcomes :-
Students will understand how to utilise backtracking.

S/W & H/W :-
1) Python3
2) 64 bit OS

Theory :-

Backtracking :-
- It is a general algo. for finding all some computational problems, notably, constraint satisfaction problem that incrementally builds candidateds to the solution.

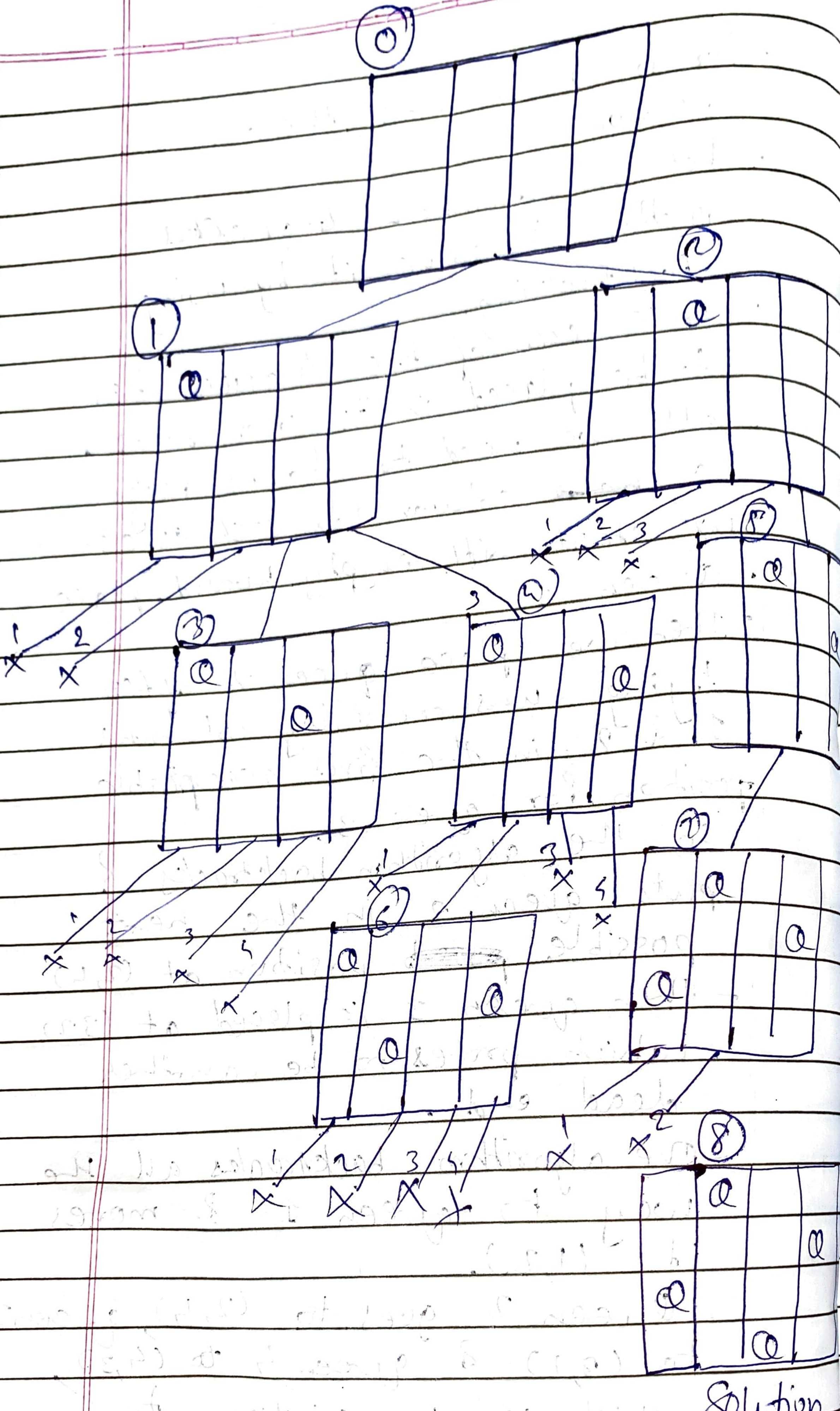
PTO

- It can only be applied for problems which admit the concept of a "partial candidate solution" & as relatively quick test of whether it can possibley be completed to a valid solution.
- It is convinient to implement this kind of processing by constructing a tree of choices being made, called the state space tree.
- It is called non-promising leaves represent other non promising dead ends or complete solutions found by the algo.

N-Queens problem:-

- The problem is to place n queens on a nxn chessboard so that no two queens attack each other by being in the same row or in the same column or on the same diagnol.
- For n=1, the problem has a trivial solution.

- for n= 2 & n=3, there is no solution.
- Let us consider 4-queens problem & solve it by using backtracking.
- Since each of the 4-queen has to be placed in its own rows, all we need to do is assign a column for each queen.
- we start with empty board, place Queen 1.
- Then we place queen 2 after trying unsuccessfully columns 1 & 2, in the first acceptable position for queen 3.
- So, the algorithm backtracks & puts queen 2 in the next possible position at (2,4).
- Then queen 3 is placed at (3,2) which proves to be another dead end.
- The algorithm backtracks all the way to queen 1 & moves it (1,2).
- Queen 2 goes to (2,4), queen 3 to (3,1) & queen 4 to (4,3), which is the solution to the problem.



Solution

Test case:-

Description	Expected	Actual	Result
1) Enter size of board	Size of board is input	Size of board is input	Success
2) If puzzle is solvable	Program terminate with soln.	Program terminate with soln.	Success
3) Unsolvable instance	Prints "no soln."	Prints "no soln."	Success

Conclusion:- Successfully implement n-queens problem.

—— x ——→ x ——