PAGE NO.:
DATE:

Name - Shweta Singh
Class - CS-3
Roll no. - 0818CS181134

Q.2

Backtracking is the most general technique to solve many problems that deal with searching for a set of solutions or which ask for an optimal solution satisfying some constraints. The name backtracking was first coined by D.H. Lehmer in the 1950s

In many applications of the back-track method, the desired solution is expressible as an n-tuple $(x_1, x_2, .., x_n)$, where the x_i are chosen from some finite set S_i. Generally the problem to be solved calls for finding one vector that maximizes (or minimizes or satisfies) a criterion function $P(x_1, x_2, ..., x_n)$. Sometimes it seeks all vectors that satisfy P.

(i) Sorting the array of integers in $a\{1:n\}$
(ii) 8-queens problem

PAGE NO.
DATE

(iii) 4-queens problem, or in generalised way n queens problem
(iv) Sum of subsets problem.

The n-queens problem is a generalised problem of 8-queens or 4-queens prob. Here, we can think that there are n queens to be placed on a n x n. Chess-board. That means, we have a chess board such that having n rows and n columns, and n queens are to be placed on this n x n Chess board i.e no two queens are in same row or in same column or in same diagonal. i.e no two queens attack each other.

Here, we suppose that queen i is to be placed in row i, say, 1 queen will be placed in first row only, but can have any column from 1, 2, ..., n so that satisfy the explicit and implicit constraints.

All Solutions to the n-queens problem can therefore be represented as n-tuples $(x_1, x_2, x_3, x_4 \ldots x_n)$ where x_i is the column on which queen i is placed. The explicit constraints using this formulation are $S_i = \{1, 2, 3, \ldots n-1, n\}$, where $1 \leq i \leq n$. ∴, the solution space consists of n^n n-tuples. Now, considering the implicit constraints that no two x_i's can be the same i.e., two queens can not be in same row, same column, or in same diagonal. So, each x_i should be different. So, by above constraints our solution space can be reduced as all solutions are permutations of the n-tuple $(1, 2, 3, \ldots n-1, n)$. So by this constraint, our solution space reduces from n^n tuples to $n!$ tuples.

Based on above constraints using backtracking technique, we can solve our problem.

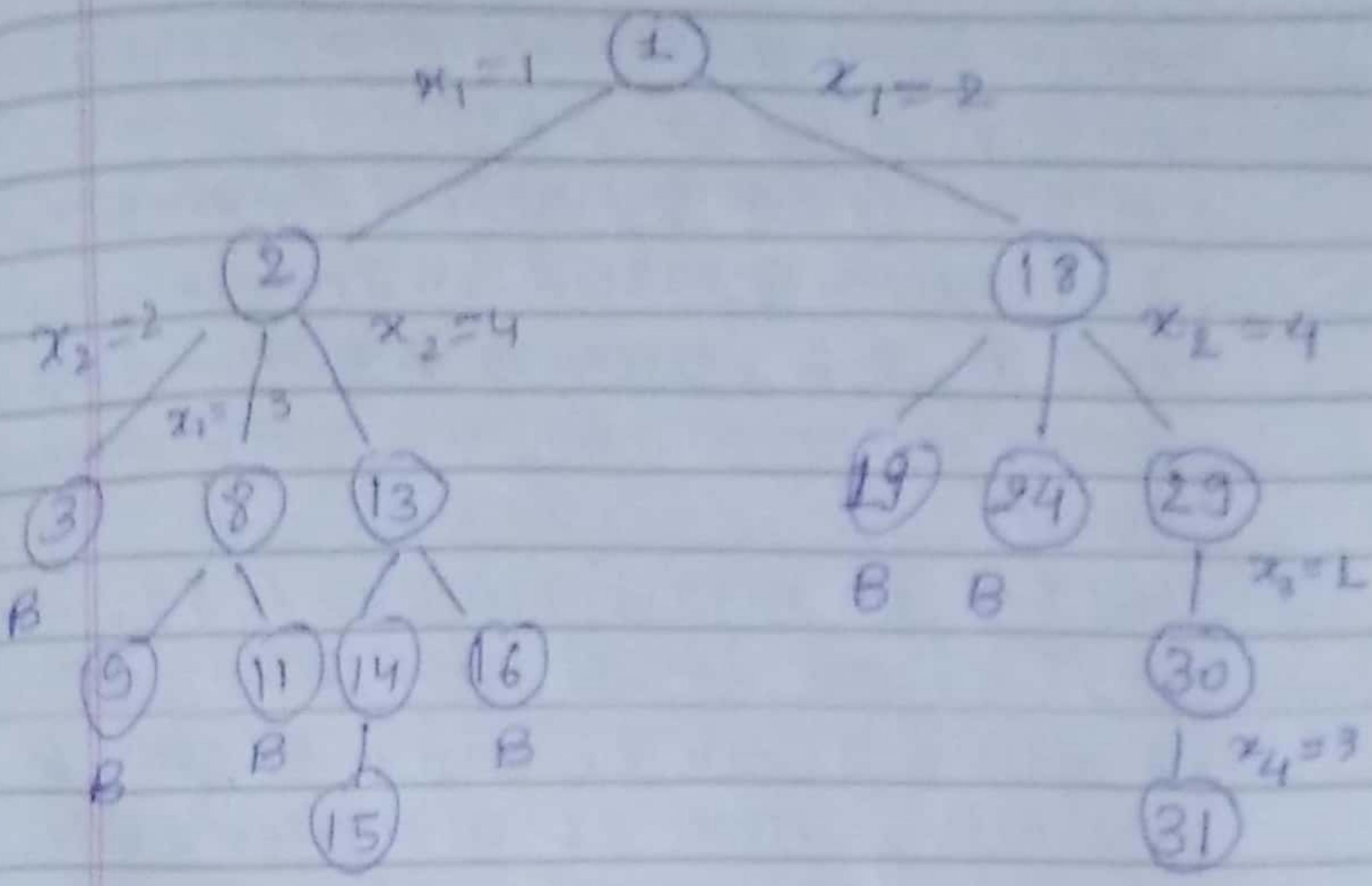
Backtracking method can be applied on 4-queens problem to solve it. In this technique, as a bounding funct-ion, we use the obvious criterion that if $(x_1, x_2, \ldots x_i)$ is the path to the current E-node, then all the children nodes with parent-child labeling x_{i+1} are such that-

$(x_1, x_2, \ldots, x_{i+1})$ represents a chessboard configuration in which no two queens are attacking.

We start with the root node as the only live node. This time this node becomes the E-node and the path is (). We generate the next child. Suppose we are generating the child in ascending order. Thus, node number 2 of fig. is generated and the path is now 1.

This means that the queen 1 is placed in first row and in first column as shown.

Now, node 2 becomes the next E node or live node. Further, try next



ascending node that node is 8 having $x_2 = 2$, i.e., queen 2 placed in 2nd column but if we do this then queen 1 & queen 2 become in same diagonal. So node 3 becomes dead here, and we backtrack to node 2 & try next possi-ble node, i.e., node 8.

Here, $x_2 = 3$, i.e., queen 2 is placed in 3rd column as shown.

This satisfy all the constraints, so,

now, node 8 becomes the next live node.

After this we try node 9 having $x_3 = 2$ i.e., queen 3 placed in 2nd column, but by this, queen 2 & queen 3 are in same diagonal. So, this node becomes dead.

We try for next possible node 11, with $x_3 = 4$. But in this case also queen 2 & queen 3 are in same diagonal resulting node 11 as a dead node.

We have tried all possible positions for queen 3, i.e., column 1, 2, 3, 4, but any of positions is not satisfying all the constraints shown so we backtrack to previous live node, i.e., node 2 & try another possible node. i.e node 13.

Now, node 13 becomes the new live node with $x_2 = 4$, i.e. queen 2 is placed in 4^{th} column as shown

After this we try next node, i.e., node 14. It becomes next live node

PAGE NO.:
DATE:

with $x_3 = 2$, i.e., queen 3 is placed in 2nd column as shown. Further we go ahead with node 15 as new live node with $x_4 = 3$. But this makes queen 3 & queen 4 placed in same diagonal resulting in a dead node 15. & we backtrack to node 14, and then backtrack to node 13, & try node 16. with $x_3 = 3$ but this results in queen 2 and queen 3 is same diagonal. So this node also becomes dead

We further back track to node 2 but no other node is left to try, so it is killed & we move back to node 1. i.e queen 1 placed in 2nd column

Again nodes 19 & 24 are killed & we try node 29 with $x_2 = 4$, i.e, queen 2 placed in 4th column as shown. Then we try node 30 as next live node with $x_3 = 1$ & finally proceed to node 31 with $x_4 = 3$ i.e queen 4 placed in 3rd column.

there for, all constraints are Satisfied and we have ~~desid~~ desired result {2, 4, 1, 3} for 4 queen problem.

	1		
			2
3			
		4	