Backtracking

Q.1 Explain backtracking approach

⇒ i) Backtracking is a process where the entire problem is divided into several stages.

ii) The algorithm then attempts to find the solution to the problem by constructing partial solutions remain consistent with requirement of the problem.

iii) However, when an incosistency with requirement occurs the problem backs up by removing the most recently constructed part of the solution and trying another possibility.

For eg., N-Queen, Graph coloring.

1) N-Queen

⇒ i) The N-Queen problem is a classic puzzle in which the task is to place N Queens on an N×N chessboard such that no two queens can attack on each other.

ii) In other words, no two queens should share the same row, column, or diagonal.

iii) Problem : Given 4×4 chessboard, arrange four queens in a way, such that no two queens attack on each other.

	1	2	3	4
1				
2				
3				
4				

- We have arrange Four Queens, Q_1, Q_2, Q_3 and Q_4 in 4×4 chess board.
- Let us start with position (1,1). Q_1 is the only queen, so there is no issue. partial solution is <1>.
- We cannot place Q_2 at (2,1) and (2,2). Position (2,3) is acceptable. partial solution is <1, 3>.
- Next Q_3 cannot be placed in position (3,1), (3,2), (3,3) or (3,4) as Q_2 and Q_1

attacks her. There is no way to put Q_3 in third row.

- Hence, this algorithm backtracks and goes back to the previous solution and readjust the positions of queen Q_2. Q_2 is moved from (2,3) to (2,4). The partial solution is <1,4>.
- Now Q_3 can be placed at position (3,2). Partial solution <1,4,3>
- Queen Q_4 cannot be placed in anywhere in row four. So again, backtrack to the previous solution and readjust the position of Q_3. Q_3 cannot be placed on (3,3) or (3,4). So the algorithm backtracks even further.
- All possible choices for Q_2 are already explored, hence algorithm goes back to partial solution <1> and moves queen Q_1 from (1,1) to (1,2).
  And this process continues till the solution is found.
- All possible solutions for 4-queen

⇒

	1	2	3	4
1		Q_1		
2				Q_2
3	Q_3			
4			Q_4	

	1	2	3	4
1			Q_1	
2	Q_2			
3				Q_3
4		Q_4		

- <u>Algorithm</u>:

```
NQUEEN (k, n)
{
    for i ← 1 to n do
        if PLACE (k, i) then
            m[k] ← i

            if k == n
                print x[1.....n]
            else
                NQUEEN ( k+1, n)
}

PLACE (k, i)
{
    for j ← 1 to k-1 do
        if x[j] == i OR (abs (x[j] - i) == abs(j-k))
            return false

    return true
}
```

- Time Complexity : $T(n) = O(n!)$

classmate

Date

Page

Q.3 Sum of subset

⇒

- SUMOFSUBSET (i, sum, W, remSum)

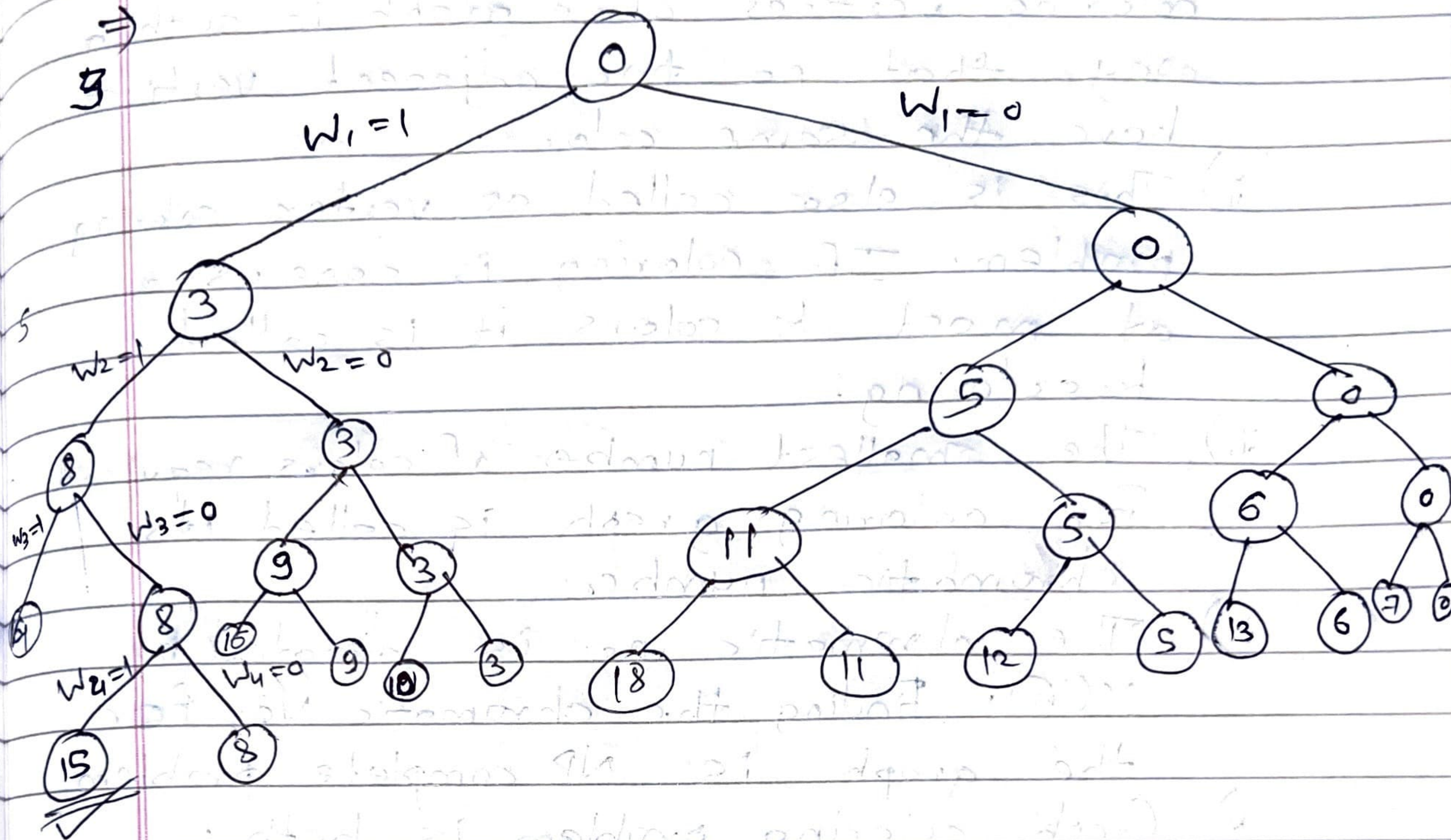
```
{
    if FEASIBLE (i) == 1 then
        if (sum == W)
            print x[1....i]

    else
        x[i+1] ← 1
        SUMOFSUBSET( i+1, sum+w[i]+1, W,
                          remsum - w[i]+1)

        x[i+1] ← 0
        SUMOFSUBSET (i+1, sum, W, remSum - w[i]+1)
}
```

```
Function FEASIBLE (i)
{
    if (sum+ remsum ≥ W) AND
    (sum == W) or sum+w[i] +1 ≤ W) then

        return 0

    return
}
```

For e.g.,

$n = 4$, $M = 15$, $W = \{3, 5, 6, 7\}$

⇒



Subset ⇒ $\{3, 5, 7\}$

Q.4 Graph Coloring.

⇒ i) Graph coloring is the problem of coloring vertices of a graph in such a way that no two adjacent vertices have the same color.

ii) This is also called as vertex coloring problem. If coloring is done using at most k colors, it is called k-coloring.

iii) The smallest number of colors required for coloring graph is called its chromatic number.

iv) The chromatic no. is denoted by $X(G)$. Finding the chromatic No. for the graph is NP complete problem.

v) Graph coloring problem is both, decision problem as well as optimization problem.

vi) The decision problem is stated as " With given M colors and Graph G, whether such color scheme is possible or not".

vii) The optimization problem is stated as " Given M colors and graph G, find the minimum no. of colors required for graph coloring".

viii) This problem can be solved using backtracking algorithm as follows:

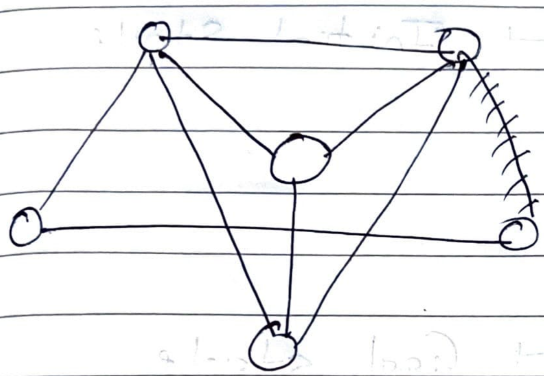
a) List down all vertices and colors in two lists.

b) Assign color 1 to vertex 1.

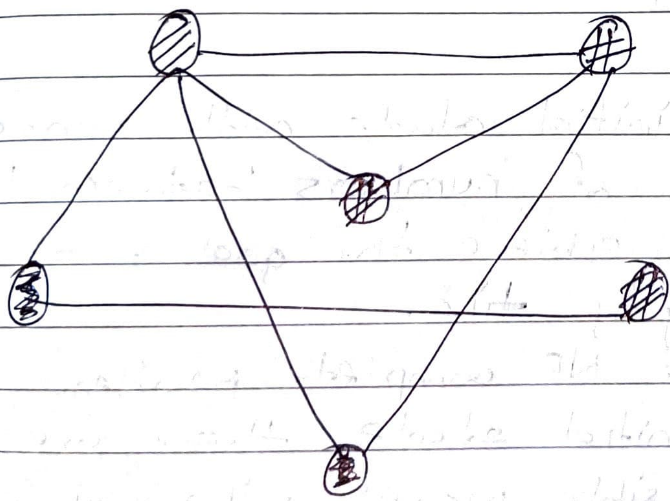
c) If vertex 2 is not adjacent to vertex 1 assign the same colors, else assign color 2.

d) Repeat the process untill all vertices are colored.

- Example :



→ non-colored graph



colored graph

Q.5 15-puzzle problem

⇒ i) 15-puzzle problem is the problem of arranging 15 titles in 4×4 board such that titles are ordered from left to right and top to bottom such that bottom right contains empty space.

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

→ Initial State

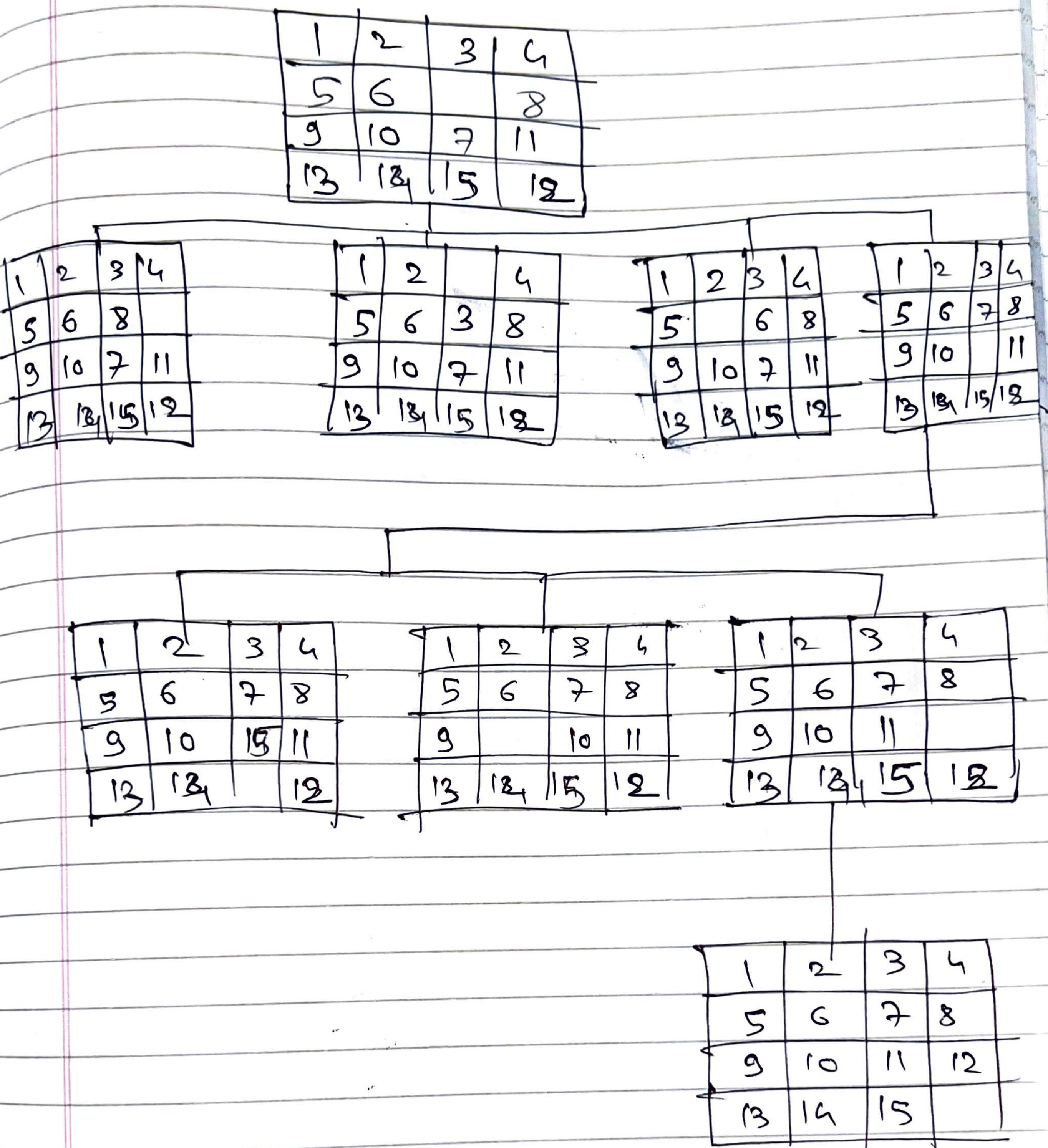
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

→ Goal state

ii) Given the initial state with random distribution of numbers between 1 to 15 aim is to achieve the goal state by moving empty tile.

iii) This is the NP complete problem.

iv) For the initial state, there are four possible moves, the cost for all four ~~cong~~ configuration is shown below -


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