MODULE - 5

BACKTRACKING → DFS ①
BRANCH AND BOUND → BFS

~~Backt~~

BACKTRACKING → (DFS) When we discover that we are going wrong in our path, we change our direction, using recursion.

Types :-
i) Decision → yes (or) no, grid
ii) optimization → easiest
iii) Enumeration → total paths

~~N-QUEENS~~ Brute force approach → try out all types of solution ~~and~~, but pick up one desired solution.

Exponential time complexity

* N-Queens
* Sum of subset
* Hamiltonian
* Knapsack
* Graph coloring

Eg :- 3 students

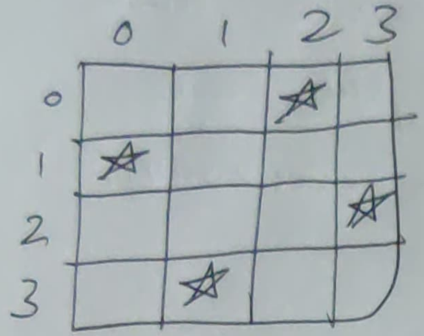
B1 B2 G1

n=3

3! = 6 solu

N-QUEENS

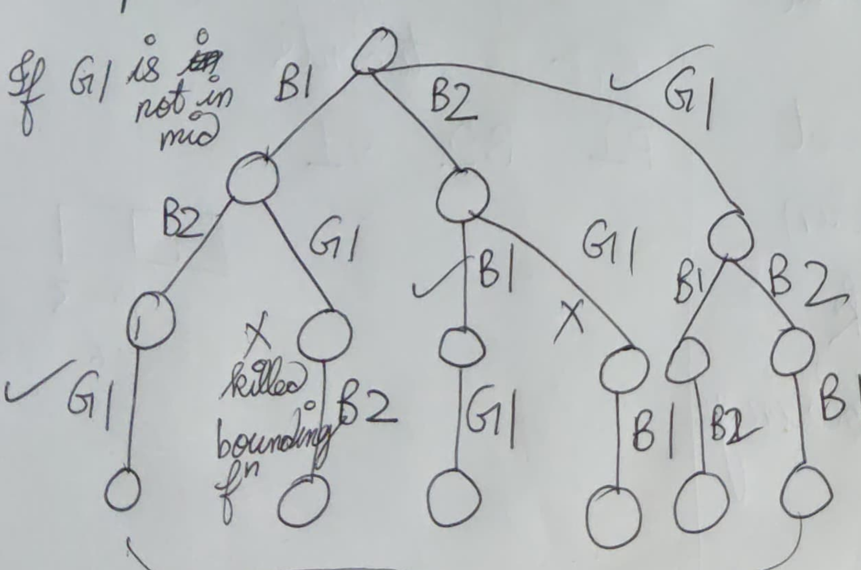
Place N queens on an N×N chessboard such that no 2 queens can attack each other



i) All solutions ⇒ char [ ][ ] 'Q' ✓ '.' 'x' ✗
ii) Possible solutions
iii) Count solutions

Under attack when they are in the same ~~st~~ row, column or diagonals.

All possible solutions → State Space Tree



6 solutions

B1 B2 G1

B1	B2	G1

remove G1 ✗
remove G1, B2 ✓

B1	G1	B2

remove B2 ✗
remove B2, G1 ✗
remove B1, G1, B2 ✓

B2	B1	G1

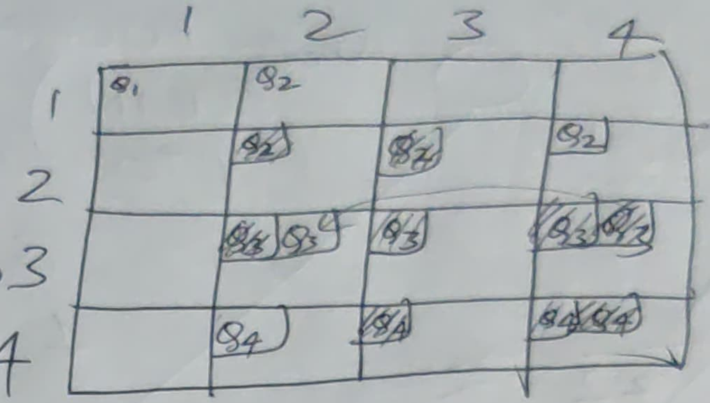
remove G1 ✗
remove G1, B1 ✓

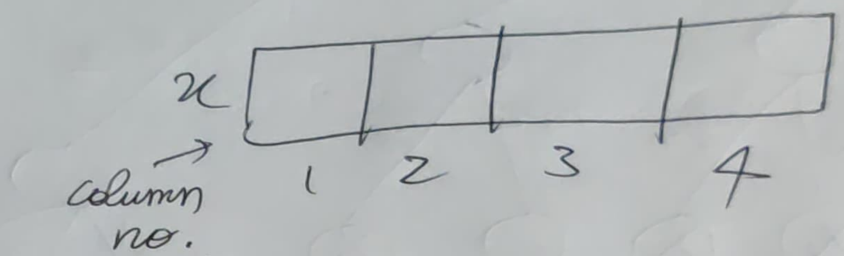
→ 4 optimised solution

N-QUEEN'S SOLVING

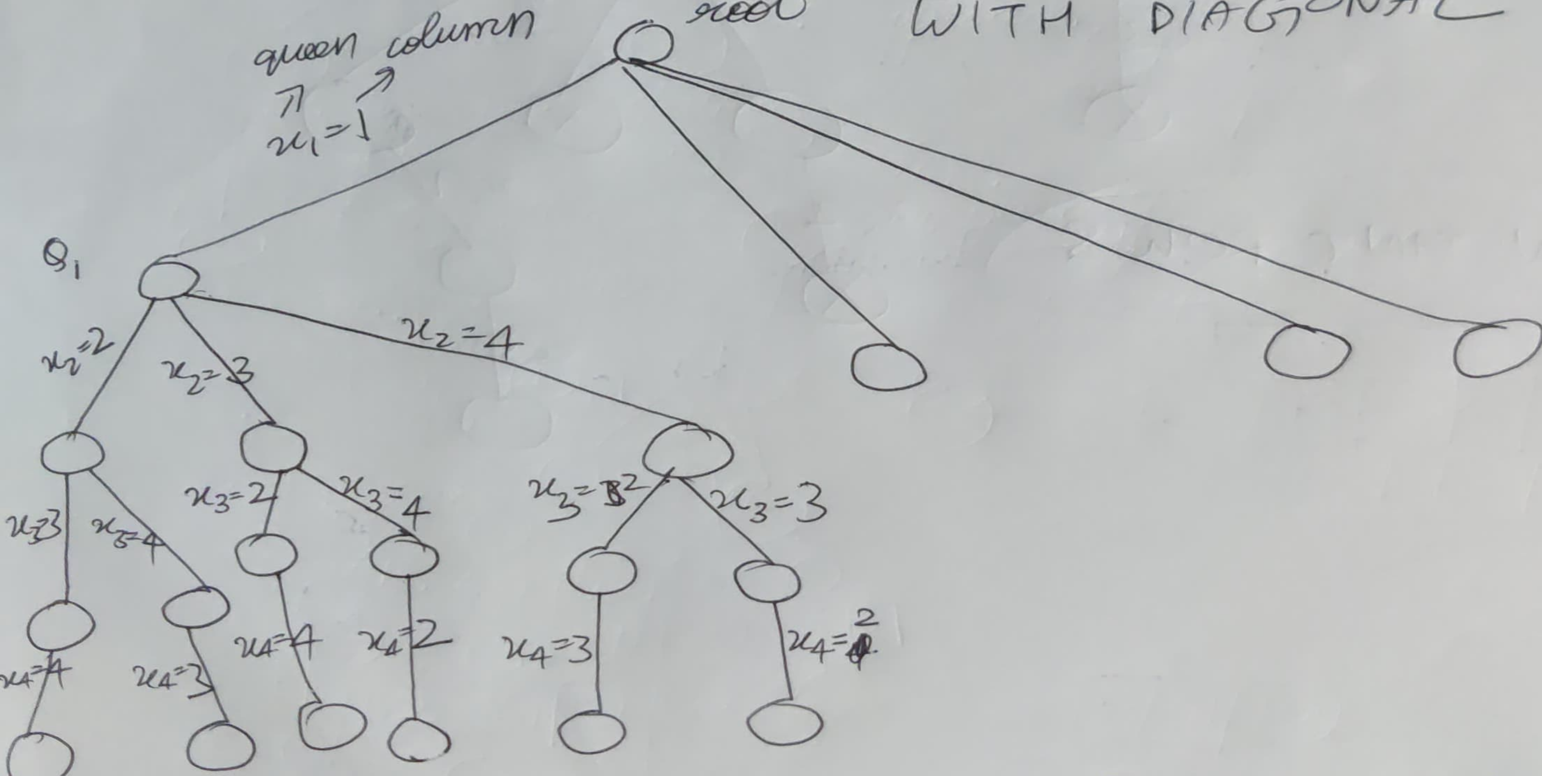
$^{16}C_4 = 1820$ ways

No same columns
row
column
diagonal



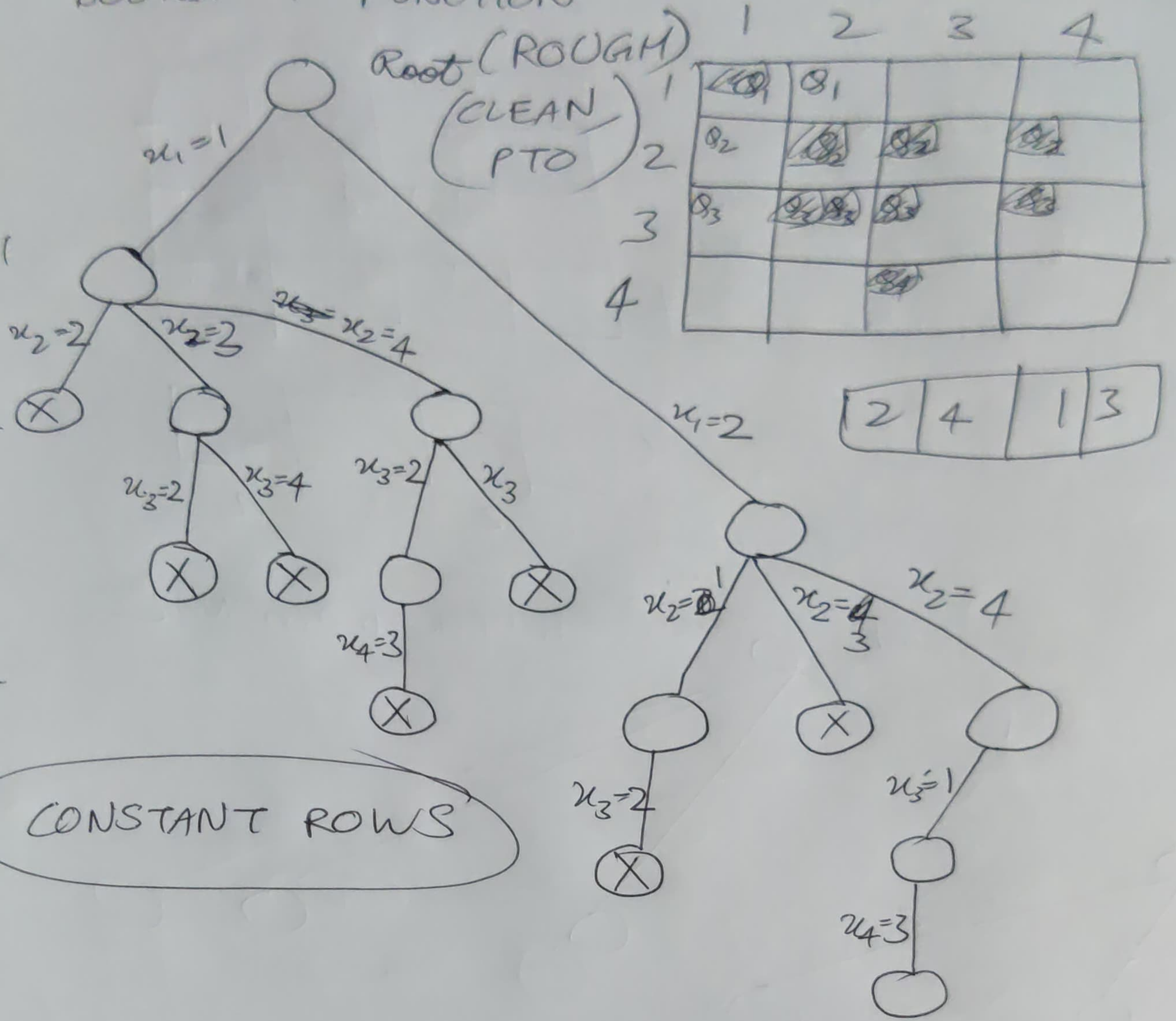


WITH DIAGONAL

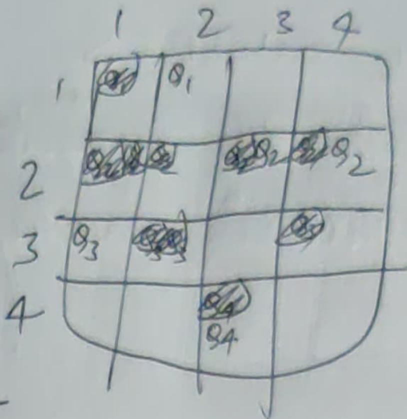


$1 + 4 + 2\times 3 + 4\times 3\times 2 + 4\times 3\times 2\times 1)$
$= 65$

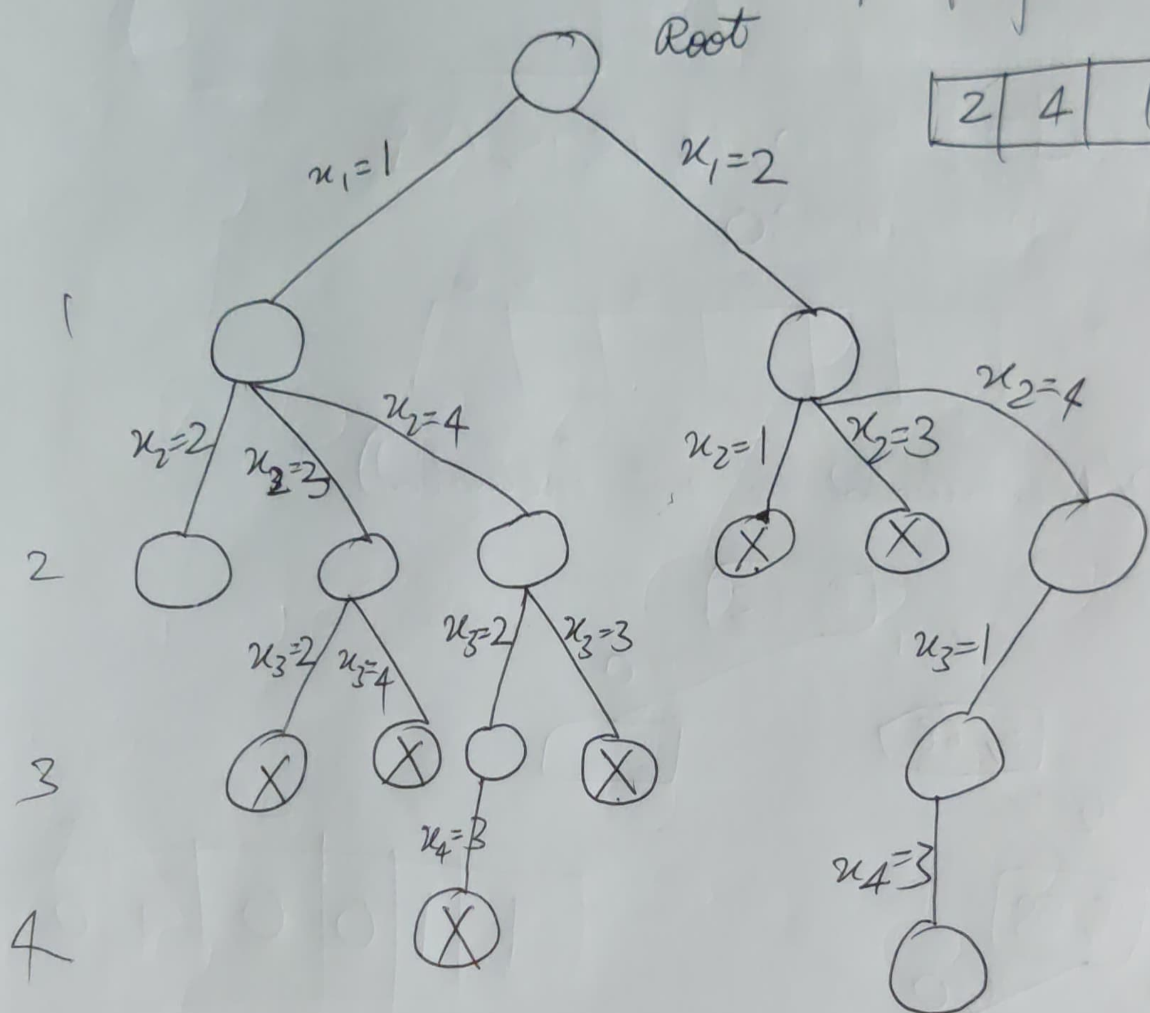
BOUNDING FUNCTION



SAME PROBLEM (CLEANER)



2	4	1	3



SUM OF SUBSETS

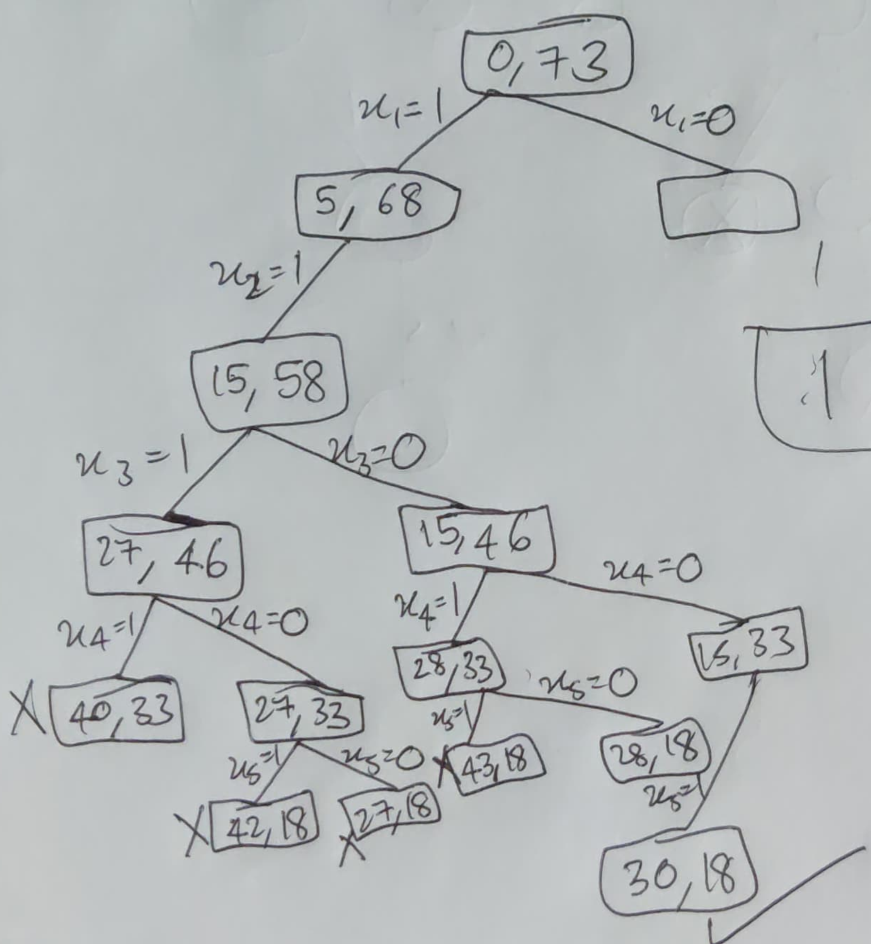
→ Take the weights such that their weight is exactly equal to m.

$w[1:6] = \{ \overset{1}{5}, \overset{2}{10}, \overset{3}{12}, \overset{4}{13}, \overset{5}{15}, \overset{6}{18} \}$

$n = 6, m = 30$

	1	2	3	4	5	6
x						

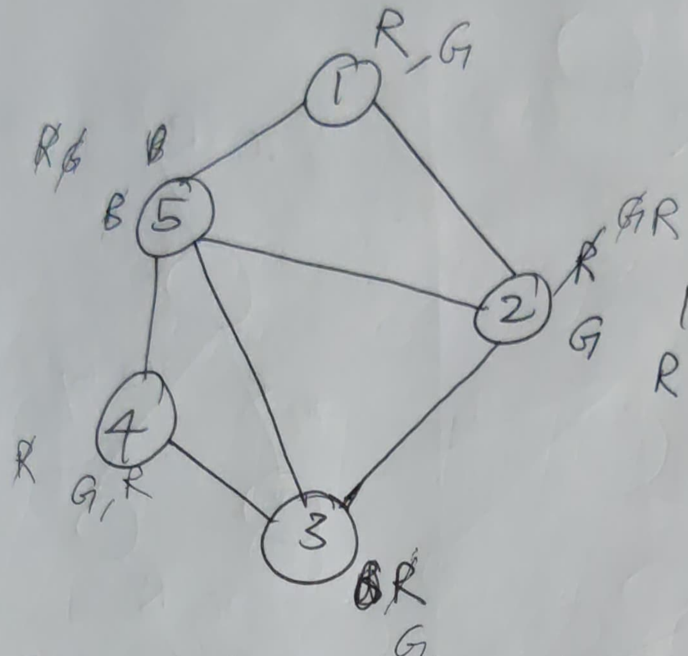
$x_i = 0/1$ (included or not included)



1	2	3	4	5	6
1	1	0	0	1	0

GRAPH COLORING PROBLEM → $\boxed{C^{n+1}}$

Color the graph such that no 2 vertices must have the same colour.



$m = 3$

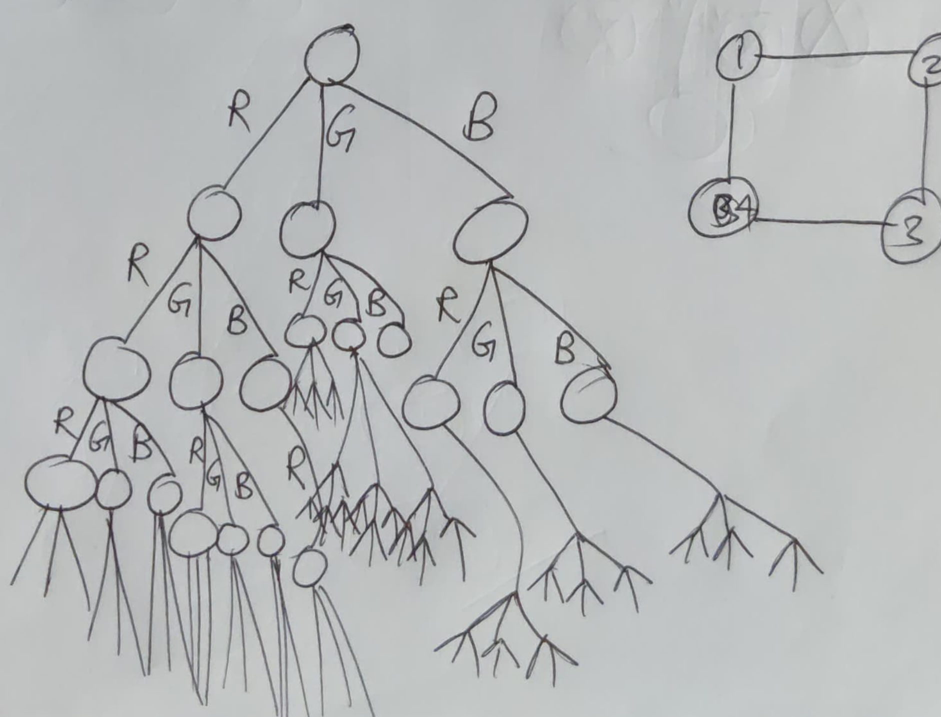
RGB

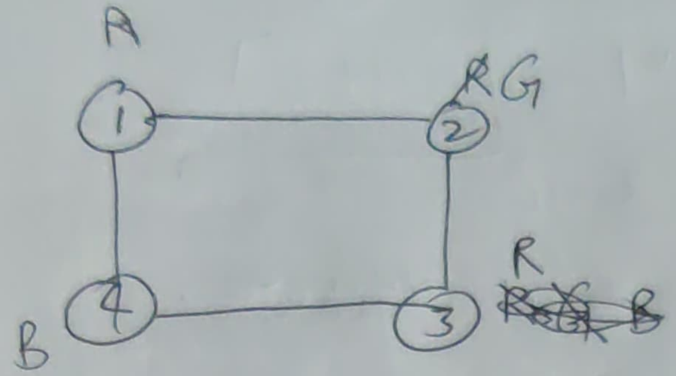
1	2	3	4	5
R	G	R	G	B

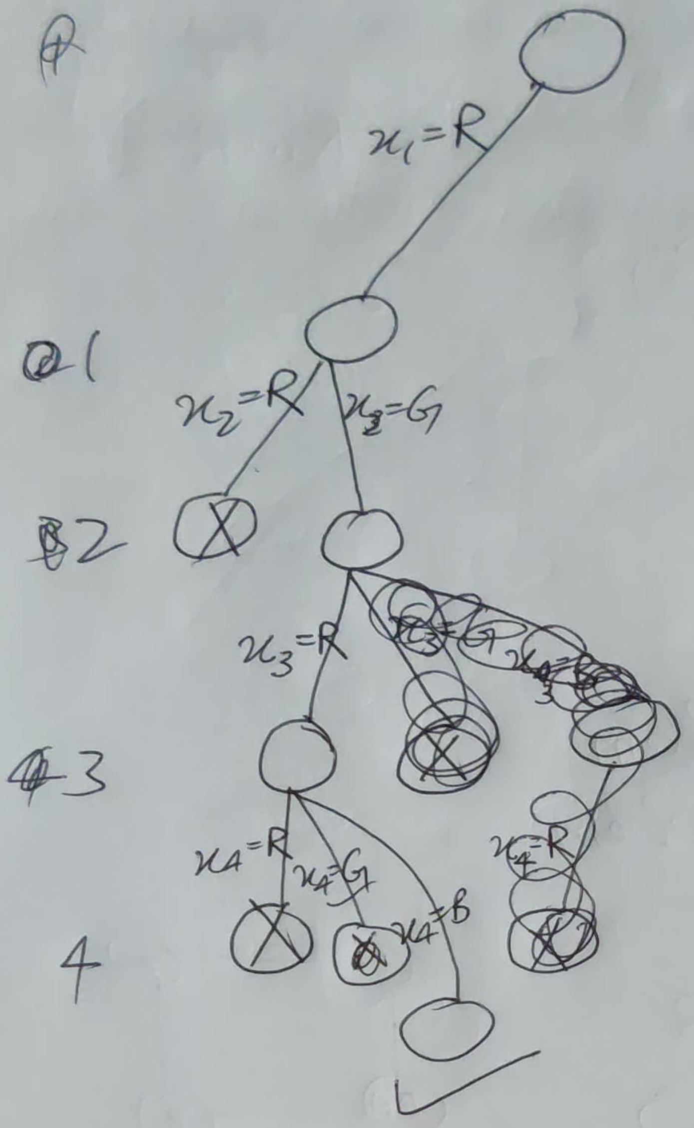
All possible solutions

m-coloring decision problem

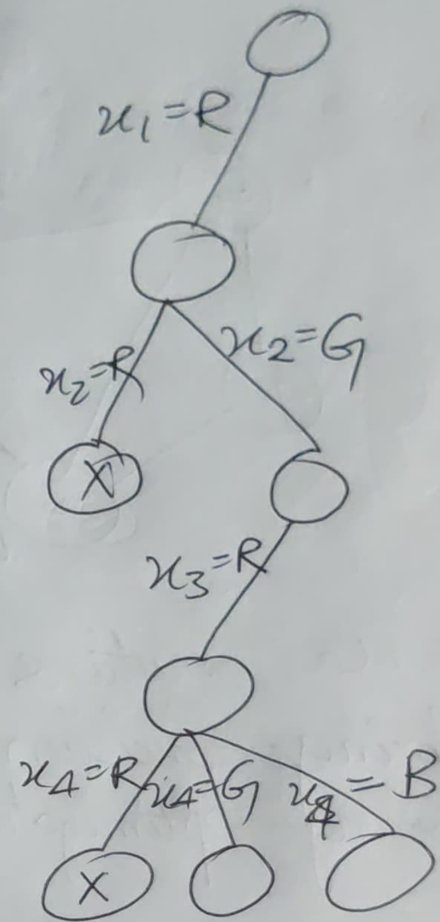
m-coloring optimization problem




A
1
2
G
R
3
4
B


$x_1=R$
1
$x_2=R$
$x_2=G$
2
$x_3=R$
3
$x_4=R$
$x_4=G$
$x_4=B$
$x_4=R$
4

Re:


$x_1=R$
$x_2=R$
$x_2=G$
$x_3=R$
$x_4=R$
$x_4=G$
$x_4=B$

P CLASS PROBLEM → DETERMINISTIC / TRACTABLE

A problem which can be solved in polynomial time by deterministic algorithms.

Eg:- ~~Hamiltom~~ Hamiltonian circuit problem
TSP
Knapsack problem
Sorting and searching

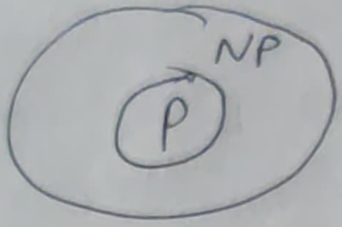
Solved and verified in polynomial time.
Easy to solve and Easy to verify.
POLYNOMIAL TIME

NP CLASS PROBLEM → NON-DETERMINISTIC / INTRACTABLE

A problem which cannot be solved in polynomial time but it is verified in polynomial time by non-deterministic algorithms → ~~NON DETE~~

Eg:- Su-Do-Ku
Prime Factor
Scheduling

Hard to solve and easy to verify
~~Verf~~ ~~Vif~~ Verified in polynomial time.
~~EXPONENENT~~
EXPONENTIAL TIME



$P \subseteq NP$

If $P = NP$,
Information security is vulnerable to ~~attack~~
Everything becomes more efficient such as ~~to~~ transportation, scheduling, etc.

If $P \neq NP$,
There are some problems which can never be solved.

~~NP HARD PROBLEM~~

REDUCTION

$$(A) \xrightarrow[\text{polynomial time}]{\text{Reduce}} (B)$$

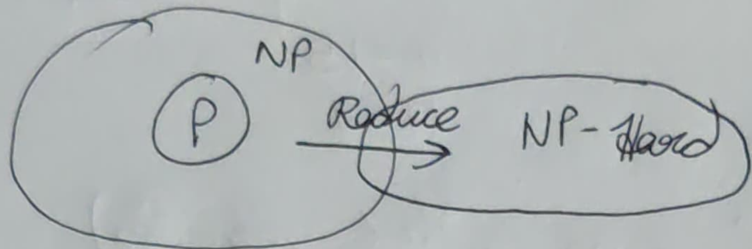
Let A and B be 2 problems,
if A reduces to B iff there is a way to solve A by deterministic algorithm in polynomial time.

$A \propto B$

i) If A is reducible to B and B in P, then A in
ii) If A is not in P $\Rightarrow$ B is not in P.

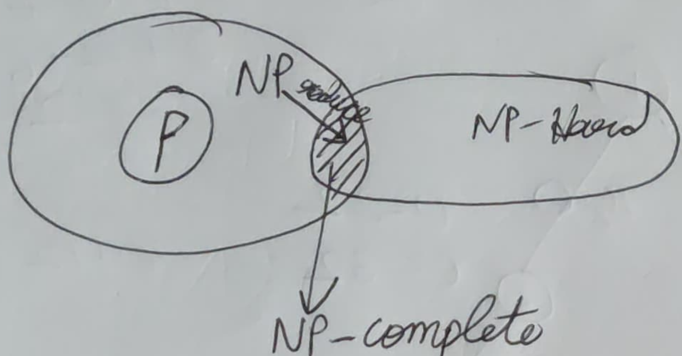
NP HARD PROBLEM → OPTIMIZATION

A problem is NP-Hard if every problem in NP can be polynomially reduced to it



NP COMPLETE PROBLEM → DECISION

A problem is NP-complete if it is in NP and it is NP-Hard



Intersection of NP and NP-hard class.

* All NP-complete problems are NP-hard but all NP-hard problems are not NP-complete.

Computational complexity
- P-class
- NP-class
  - NP-hard
  - NP-complete.

0/1 KNAPSACK USING LC BRANCH AND BOUND

Max $\xrightarrow{neg}$ Min

- Maximization problem
  Negative profit.

	1	2	3	4
profit	10	10	12	18
weight	2	4	6	9

$m = 15$

$n = 4$

LC-BB → lowest node

upper node $= \sum_{i=1}^{n} p_i x_i \le m$ (without fraction)

cost $= \sum_{i=1}^{n} p_i x_i$ (with fraction)

$S = \{x_1, x_3\}$ } both are correct

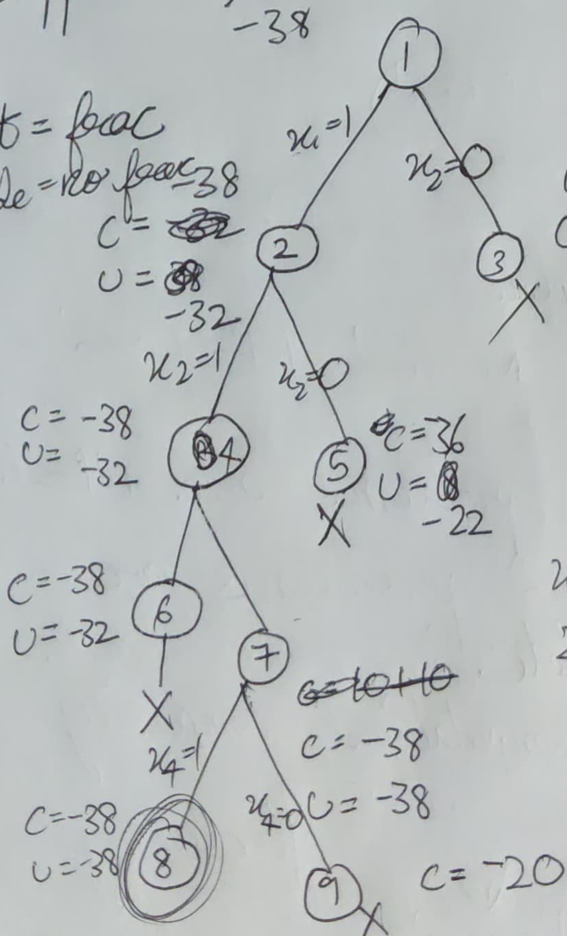
$S = \{1, 0, 1, 0\}$ → fixed size solution

$m = 15$
$n = 4$

upper $= \infty = \cancel{-32}$ $(-10-10-12)$
-38

cost $10 + 10 + 12 + \frac{18}{9}(\frac{3}{9})$
$2 + 4 + 6$
$= 12$

cost = frac
node = no frac



$10 + 12 + \frac{18}{9} \times 5$

$c = 10 + 10 + 12 + \frac{18}{9} \times \frac{7}{9}$

$U = \cancel{6}$

$2 + 6 = 8$

$x_1 + x_2 + x_4$

$2 + 4 + 9 = 15$

$10 + 10 + 18 = 38$

$c = 10 + 10 + 18$

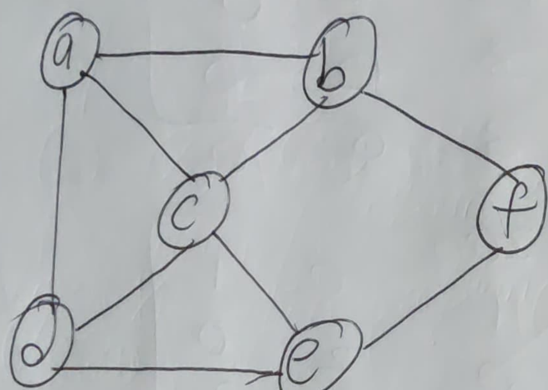
$= 2 \quad 4 \quad 9$

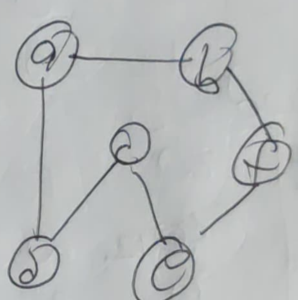
$= 15$

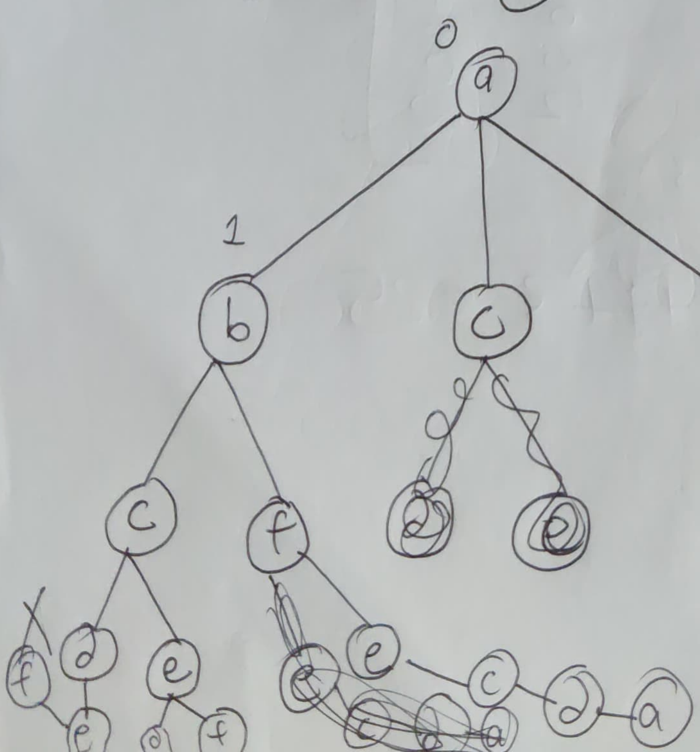
HAMILTONIAN CIRCUIT PROBLEM USING BACKTRACKING

This problem is concerned about finding Hamiltonian circuit in a given graph

Hamiltonian circuit is defined as a cycle that passes all the vertices of graph exactly once except the starting and ending vertices (same vertex)







start and come back to a-

TRAVELLING SALESPERSON PROBLEM
BRANCH AND BOUND

$$
\begin{array}{c}
\\ 1 \\ 2 \\ 3 \\ 4 \\ 5
\end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{bmatrix}
\infty & 20 & 30 & 10 & 11 \\
15 & \infty & 16 & 4 & 2 \\
3 & 5 & \infty & 2 & 4 \\
19 & 6 & 18 & \infty & 3 \\
16 & 4 & 7 & 16 & \infty
\end{bmatrix}
\end{array}
\begin{array}{c}
\text{Min} \\ 10 \\ 2 \\ 2 \\ 3 \\ 4 \\ \hline 21
\end{array}
$$

$$
\begin{bmatrix}
\infty & 10 & 20 & 0 & 1 \\
13 & \infty & 14 & 2 & 0 \\
1 & 3 & \infty & 0 & 2 \\
16 & 3 & 15 & \infty & 0 \\
12 & 0 & 3 & 12 & \infty
\end{bmatrix}
$$
$$
\begin{array}{ccccc} 1 & 0 & 3 & 0 & 0 \end{array}
$$

Reduced = atleast 1 element is 0 shortest list.

$$
\begin{bmatrix}
\infty & 10 & \cancel{20}\,17 & 0 & 1 \\
\cancel{13}\,12 & \infty & 11 & 2 & 0 \\
0 & 3 & \infty & 0 & 2 \\
15 & 3 & 12 & \infty & 0 \\
11 & 0 & 0 & 12 & \infty
\end{bmatrix}
$$

sub 1 sub 3

cost = 21 + 1 + 3 = 25