PAGE No.

DATE / /

→ Mod 4

4.1 : Backtracking (Gives all possible solutions)

Uses Bruteforce

Uses DFS & statespace Tree

Has bounding functions / constraints

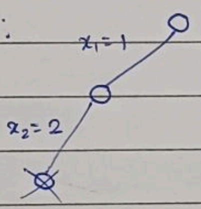
* N - Queens [$O(N!)$]

N × N matrix given & N queens need to be placed

Without diagonal constraint formula :

$1 + \sum_{i=0}^{3}\left[\prod_{j=0}^{i}(N-j)\right]$ → for total no. of possible nodes

Steps :



Go Depth wise

Backtrack if violated

Place queen at next column

Get final ans : $C_1 \to Q_1$, $C_2 \to Q_2$, $C_3 \to Q_3$, $C_4 \to Q_4$ . . .

* Sum of subsets [$O(2^n)$]

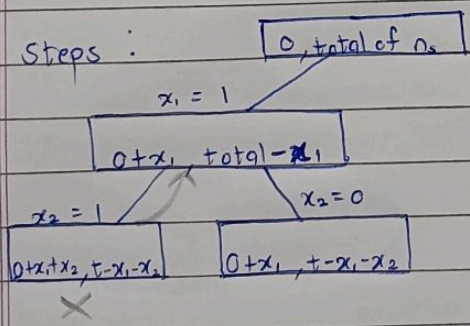
① Given some (n) weights { , , , , . . . }

② Given a sum (m) value

③ x [| | | | |||] & $x_i = 0 / 1$

1 2 3 4 . . . n

Steps :



$\sum w_i x_i + w_{k+1} \le m$

$\sum_{i=1}^{k} w_i x_i + \sum_{i=k+1}^{n} w_i \ge m$

left should be less / equal to m

& L + Right should be greater / equal to m

if Right becomes 0, stop

Write all possible answers, with their 1, 0 combo (Show for only one)

PAGE No.
DATE / /

4.2 : Branch & Bound (Used for optimization problems)

Uses BFS & Statespace Tree

Types
- → FIFO – Queue
- → LIFO – Stack
- → Least Cost

* Traveling Salesman Problem [$O(2^n) \cdot O(n^2)$]

① Will be given a graph

② Create its adjacency matrix

$$\begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} \infty & & & \\ & \infty & & \\ & & \infty & \\ & & & \infty \end{bmatrix} \end{array}$$

③ Get lowest of each row & each col

④ Subtract it from its row/col

⑤ The total of those : reduced cost (r)

⑥ Start with the BFS, start with the given point in Q.

⑦ Give each node a number along with its actual node no.

⑧ Move breadth wise finding lowest cost

Nodes : Initial (Parent) to final (Child)

Use Matrix of Parent

Make the ini row $=\infty$ & fin col $=\infty$

& also (Topmost, finalchild) $=\infty$

Check if each row has one 0
& col

If not repeat step ④ & use it as $\hat{r}$

Formula : $c(ini, fin) + r + \hat{r}$

in further stage : $c(ini, fin) + c(ini) + \hat{r}$

Repeat

PAGE No.
DATE / /

* 15 Puzzle Problem [O(N!)]

① Given a matrix with certain arrangement & a goal arrangement.
② We can move the blank space ← ↑ ↓ →
③ Count no. of errors at each of the 4
④ Expand / branch the one with least errors.
⑤ Repeat