27/8/24

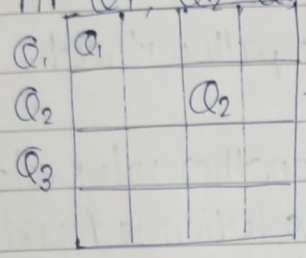
Backtracking branch and bound

☆ N-Queens:

- 4×4 Queens Matrix
- No 2 Queens should be in:
  - Same Row
  - Same Column
  - Same Diagonal

Eg: Fit Q_1, Q_2, Q_3, Q_4 in 4×4 matrix.

Q.

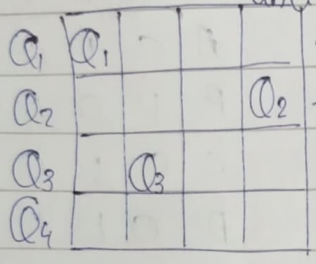


Q_1 → Pos 1

Q_2 → Pos 3

Q_3 No pos for Q_3

∴ Backtrack and change pos Q_2



Q_1 → Pos 1

Q_2 → Pos 4

Q_3 → Pos 2

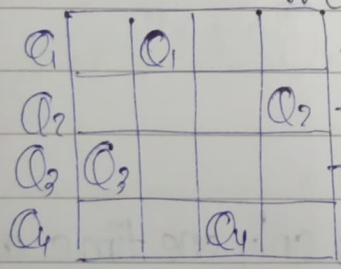
Q_4 No pos for Q_4,

∴ Afterbacktrack,

No pos for Q_3,

No pos for Q_2.

∴ Change pos Q_1.

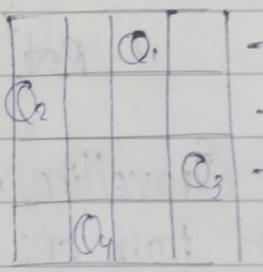


Q_1 → Pos 2

Q_2 → Pos 4

Q_3 → Pos 1

Q_4 → Pos 3 ∴ Solution 1



→ Pos 3

Q_2 → Pos 1

Q_3 → Pos 4

Q_4 ∴ Solution 2

SPCC

Draw all possibilities (in matrix) tree form.

★ Graph Colouring:

Steps: ① Select a single node (generally node 1) of a given graph and assign to it colour number 1.

② Select remaining nodes of a graph one by one & assign the higher numbered colour to them from list of available colours

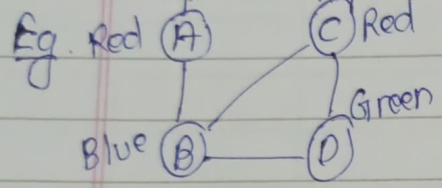
③ For each colour assignment, apply the bounding function to check 2 adjacent nodes have different colours.

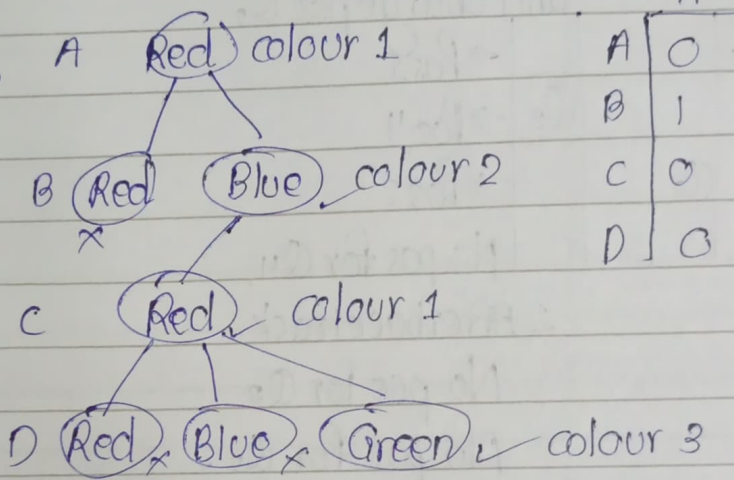
④ If partial solution is feasible discard the path without further exploration.

⑤ Back track to previous level and assign another valid colour.

⑥ Repeat steps ② to ⑤ until the complete solution is found.

Eg.





	A	B	C	D
A	0	1	0	0
B	1	0	1	1
C	0	1	0	1
D	0	1	1	0

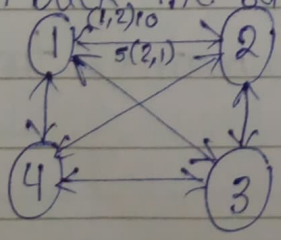
A B C D

Red Blue Red Green

★ TSP (Travelling Salesman Problem):

→ Start from one city, visit all the cities only one time and reach back the same city.

Eg.



	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0

$$g(i,S) = \min \{ C_{ik} + g(k, S-\{k\}) \}$$

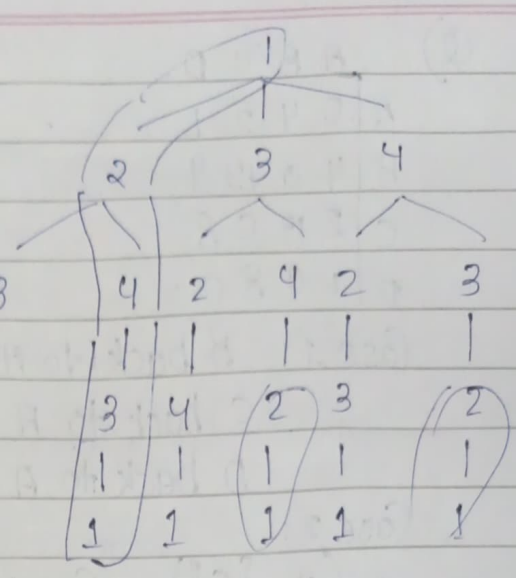
for Case 1: $C_k = g(i, \phi) = cost$

$i = 2$

2 back 1 $C_{21} = g(2, \phi) = 5$

3 back 1 $C_{31} = g(3, \phi) = 6$

4 back 1 $C_{41} = g(4, \phi) = 8$



Case 2: $g(i, k)$

$g(2, \{3\}) = C_{23} + g(3, \{3,1\} - 3) = 15$

$g(2, \{4\}) = C_{24} + g(4, \{4,1\} - 4) = 18$

$g(3, \{2\}) = C_{32} + g(2, 1) = 18$

$g(3, \{4\}) = C_{34} + g(4, 1) = 20$

$g(4, \{2\}) = C_{42} + g(2, 1) = 8 + 5 = 13$

$g(4, \{3\}) = C_{43} + g(3, 1) = 9 + 6 = 15$

Case 3: $g(2, \{3,4\}) = \min\{(C_{23} + g(3,4)), (C_{24} + g(4,3))\}$

$= \min\{(9 + 20), (10 + 15)\} = 25$

$g(3, \{2,4\}) = \min\{(C_{32} + g(2,4)), (C_{34} + g(4,2))\}$

$= \min\{(13 + 18), (12 + 13)\} = 25$

$g(4, \{2,3\}) = \min\{(C_{42} + g(2,3)), (C_{43} + g(3,2))\}$

$= \min\{(8 + 15), (9 + 18)\} = 23$

Case 4: $g(1, \{2,3,4\}) = \min\{(C_{12} + g(2, \{3,4\})), (C_{13} + g(3, \{2,4\})), (C_{14} + g(4, \{2,3\}))\}$

$= \min\{(10 + 25), (15 + 25), (20 + 23)\}$

$= 35$

$① \rightarrow ② \rightarrow ④ \rightarrow ③ \rightarrow ①$

Q)

	A	B	C	D
A	0	4	2	1
B	4	0	13	9
C	2	13	0	8
D	1	9	8	0

Case 1: B back to A = $C_{BA} = g(B, \phi) = 4$
C back to A = $C_{CA} = g(C, \phi) = 2$
D back to A = $C_{DA} = g(D, \phi) = 1$

Case 2:
$g(B, \{C\}) = C_{BC} + g(C, \phi) = 13 + 2$
$g(B, \{D\}) = C_{BD} + g(D, \phi) = 9 + 1$
$g(C, \{B\}) = C_{CB} + g(B, \phi) = 13 + 4$
$g(C, \{D\}) = C_{CD} + g(D, \phi) = 8 + 1$
$g(D, \{B\}) = C_{DB} + g(B, \phi) = 9 + 4$
$g(D, \{C\}) = C_{DC} + g(C, \phi) = 8 + 2$

Case 3:
$g(B, \{C, D\}) = \min\{(C_{BC} + g(C, \{D\})), (C_{BD} + g(D, \{C\}))\}$
$= \min\{(13 + 9), (9 + 10)\}$
$= 19$

$g(C, \{B, D\}) = \min\{(C_{CB} + g(B, \{D\})), (C_{CD} + g(D, \{B\}))\}$
$= \min\{(13 + 10), (8 + 13)\}$
$= 21$

$g(D, \{B, C\}) = \min\{(C_{DB} + g(B, \{C\})), (C_{DC} + g(C, \{B\}))\}$
$= \min\{(9 + 15), (8 + 17)\}$
$= 24$

Case 4:
$g(A, \{B, C, D\}) = \min\{(C_{AB} + g(B, \{C, D\})), (C_{AC} + g(C, \{B, D\})), (C_{AD} + g(D, \{B, C\}))\}$
$= \min\{(4 + 19), (2 + 21), (1 + 24)\}$
$= 23$

1) A → B → D → C → A
2) A → C → D → B → A

(Take base as 10 or 26 if alphabets are present)

A - A ÷ B × B

☆ String Matching:

1) Rabin Karp:

Steps: ① Check the length of the pattern given and calculate hash value as [Patt. mod q]
② Check the given text and find the hash value.
③ If the hash values match we call it as Spurious Hit.
④ From all the Spurious Hit match the characters by character to get final Match

Eg. T = 3 1 4 1 5 9 2 6 5 3 5 → Text
P = 2 6 → Pattern.
Prime no. q = 11 given (If not given take any prime no.)

→ 26 mod 11 = 4

∴
31 mod 11 = 9 ✗
14 mod 11 = 3 ✗
41 mod 11 = 8 ✗
15 mod 11 = 4 SH
59 mod 11 = 4 SH
92 mod 11 = 4 SH
26 mod 11 = 4 SH
65 mod 11 = 10 ✗
53 mod 11 = 9 ✗
35 mod 11 = 2 ✗

15, 59, 92, [26]

② T = 2 3 5 9 0 2 3 1 4 1 5 2 6 7 3 9 9
P = 3 1 4 1 5
Q = 13

→ 31415 mod 13 = 7

23590 % 13 = 8
35902 % 13 = 9
59023 % 13 = 3
90231 % 13 = 11
02314 % 13 = 0
23141 % 13 = 1
31415 % 13 = 7
14152 % 13 = 8
41526 % 13 = 4
15267 % 13 = 5
52673 % 13 = 10
26739 % 13 = 11
67399 % 13 = 7

[31415], 67399

C C A

$3\times10^2+3\times10^1+3\times10^0$

Q) T = AABCDACCAD

P = CCA

Q = 13

→ CCA % 13 = 331 % 13 = 6

112 % 13 = 8

123 % 13 = 6

234 % 13 = 0

341 % 13 = 3

413 % 13 = 10

133 % 13 = 3

331 % 13 = 6

314 % 13 = 2

331 | 123 CCA | ABC

★ Knuth Morris pattern:

→ Prepare π table for given pattern → first time write as 0 next time it appears → index no.

→ String (i) with pattern j + 1

if matching, move i & j (Pi table)

else (Mismatch), Move (j) to the below index.

creating π(Pi) Table,

1) index 1 2 3 4 5 6 7 8 9 10

a	b	c	d	a	b	e	a	b	f
0	0	0	0	1	2	0	1	2	0

2) Index 1 2 3 4 5 6 7 8 9 10 11

a	b	c	d	e	a	b	f	a	b	c
0	0	0	0	0	1	2	0	1	2	3

3) Index 1 2 3 4 5 6 7 8 9 10

a	a	b	c	a	d	a	a	b	e
0	1	0	0	1	0	1	2	3	0

4) Index 1 2 3 4 5 6 7 8 9

a	a	a	a	b	a	a	c	d
0	1	2	3	0	1	2	0	0

(Q) i

String = a b a b c a b c a b a b a b d
1 2 3 4 5

Pat = a b a b d

j → | 0 | 0 | 1 | 2 | 0 |

a=a i++ j++	a=a i++ j++	a=a i++ j++
b=b i++ j++	b=b i++ j++	b=b i++ j++
a=a i++ j++	c≠a i++ j=1	a=a i++ j++
b=b i++ j++	a≠b i++ j=0	b=b i++ j++
c≠d i++ j=0	b≠a i++ j=0	d=d i++ j++