Gas Station Problem :→

Station → i : 0 1 2 3 4

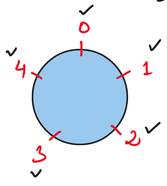
gas[i] = 1, 2, 3, 4, 5

cost[i] = 3, 4, 5, 1, 2

(Petrol Pump)

[0 ip = 3]

[0 or (+) ve] complete on circuit



Station 3: 0+4-1 = (3)

Station 4: 3+5-2 = (6)

Station 0: 6+1-3 = (4)

Station 1: 4+2-4 = (2)

Station 2: 2+3-5 = (0)

Station 3: 0+4-1 = (3)

if possible return station index (i) else return (-1).

(curr tank < 0) start + 1

0 1 2 3 4

gas = { 1, 2, 3, 4, 5 }

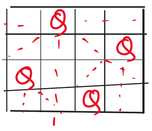
cost = { 3, 4, 5, 1, 2 }

i	gas[i]	cost[i]	diff	curr tank	total tank	start index
0	1	3	-2	-2	-2	1
1	2	4	-2	-2	-4	2
2	3	5	-2	-2	-6	3
3	4	1	+3	+3	-3	3
4	5	2	+3	+3	0	3

Back Tracking :→

4x4 Matrix 4 Queens N Queens

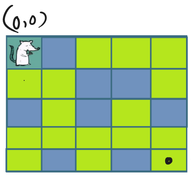
{ Q, Q, Q, Q }



[Rat In A Maze]

Rat in a Maze :→

nxn matrix



(Start) rat (0, 0) dest^n (n-1, n-1)

green - path → 1

black - walls → 0

Traverse → forward (x+1, y)

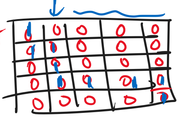
→ downward (x, y+1)

* curr pos (x, y) → (x, y) If i can stand [x, y] = [i]

for → if (x+1, y)

down → else (x, y+1)

nxn result mat



Dynamic Programming :→

* " Those who forget the past, are forced to repeat it. "
* The solution of small overlapping sub-problems to solve a bigger problem.
* ① Recursion
  ② Memoization → Top Down
  ③ Tabulation → Bottom Up
  ④ Space Optimization

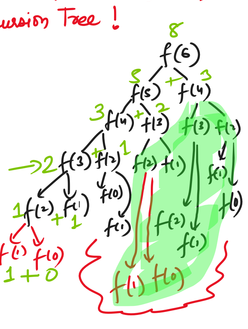
Fibonacci Series : (n^th)

0, 1, 1, 2, 3, 5, 8, 13, 21, ...

0 1 2 3 4 5 6 7 8

f(n) = f(n-1) + f(n-2)

Recursion Tree !



{ C++ → Code Help / Take U Forward
Python → Tech With Tim (Debeu)
Java → Kunal Kushwaha (striver) } Algos Abdul Bari

→ * [B+ B Tree DBMS]

[Free Code Camp]

[Chai Aur Code]

[VTU] → Coders Arcade

(Perplexity) Dry

[Prepinsta → Top 100 Codes] ***** 10 11th

↳ Leet Code

(Copilot)

[Bit Manipulations]

* Stored Procedures delimiter