

* Dynamic Pattern Problems for Competitive Coding:
(Capgemini / Cognizant / TCS)

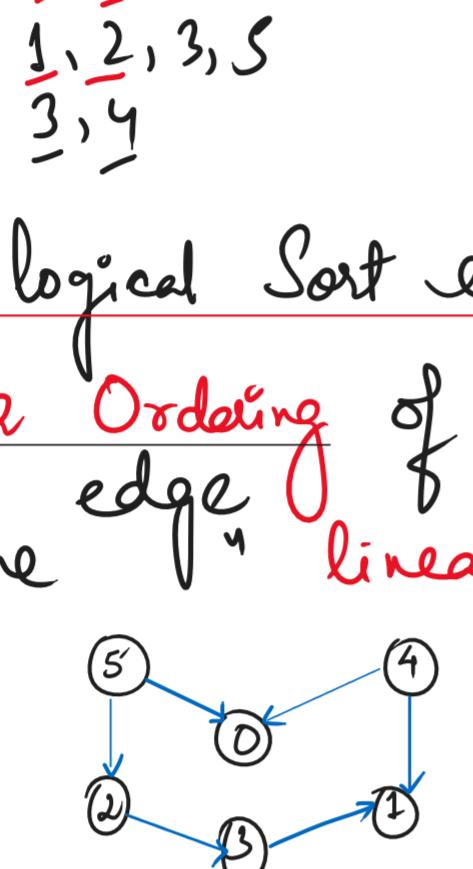
	1	2	3	4	5	6	7	8	9	10	11	12	13	Δ (Triangle)
1	*				*				*		*			set 2
2	*		*		*		*		*		*		*	1 (x+y) / 4 == 0 2 & 4 == 0

Values: row (constant) = 3

Values: col (variable) = 9, 13, 17, 21 and so on

$$\begin{aligned} \text{Time: } &\rightarrow 3 \text{ ms} \\ \text{so } \left\{ \begin{array}{l} r_1 : 3, 7, 11 \rightarrow C \cdot 4 = 3 \\ r_2 : \text{even nos} \rightarrow C \cdot 2 = 0 \\ r_3 : 1, 5, 9, 13 \rightarrow C \cdot 4 = 1 \end{array} \right. \end{aligned}$$

Graph Traversals: →



BFS (queue) → DFS (recursion)

Output: 1, 2, 3, 4, 5

→ Who are your neighbours?

Adj-list:

1 → 2, 3
2 → 1, 3, 4
3 → 1, 2, 4, 5
4 → 2, 3, 5
5 → 3, 4

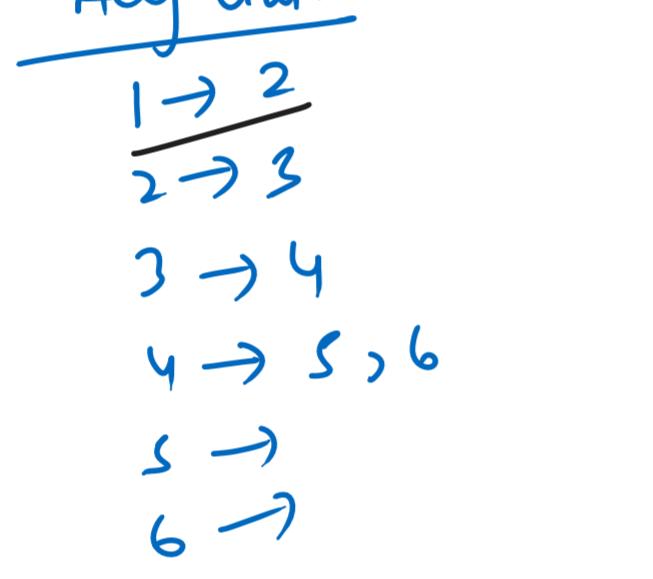
Visited

1	F	T
2	F	T
3	F	T
4	F	T
5	F	T

X	X	X
X	X	X
X	X	X

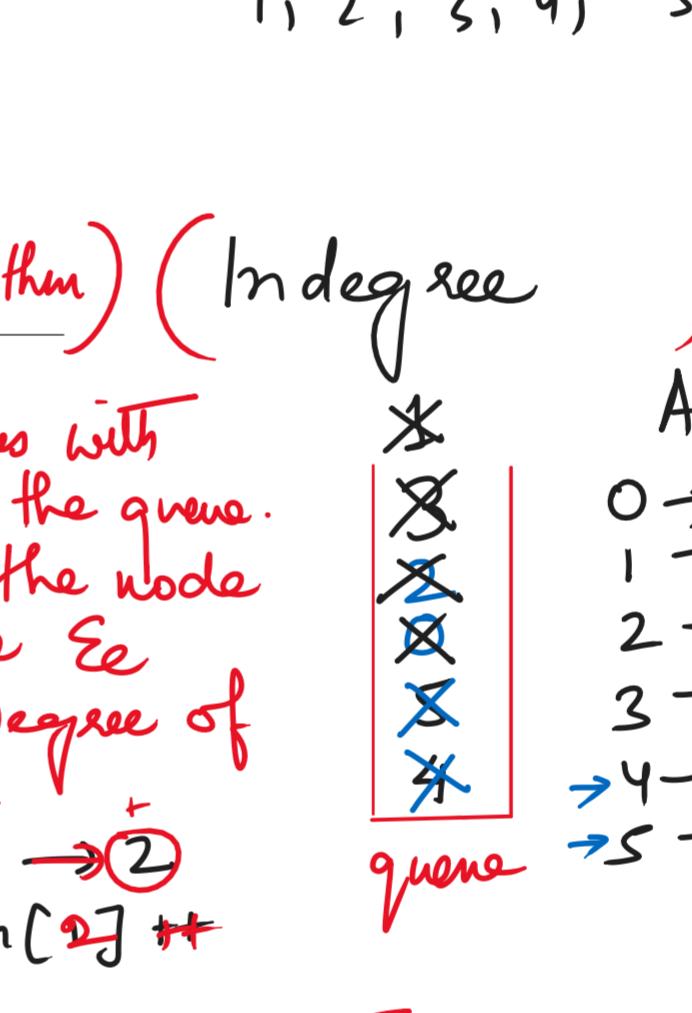
queue (FIFO)

DFS Traversal Graph: →



Note: Depending on which way you are traversing, different ops are possible! All are accepted & correct.

O/p: → 1, 2, 4, 3, 5



* Adj List:

1 → 2, 3, 4
2 → 1, 3, 5
3 → 1, 2, 4, 5
4 → 2, 3, 5
5 → 3, 4

* Visited Array:

1	F	T
2	F	T
3	F	T
4	F	T
5	F	T

Topological Sort Algorithm: → (Directed Acyclic Graph)

Linear Ordering of vertices in such a way that if there is an edge from Node "u" to Node "v", "u" always comes before "v".

Edges:

1 → 0
1 → 2
2 → 3
4 → 0
4 → 2
2 → 1

Component-wise traversal:

for (0 - s) → dfs(s)

0, 1, 2, 3, 4, 5

Adj list:

1 → 2
2 → 3
3 → 4
4 → 5
5 → 0

Visited array:

0 → F
1 → T
2 → F
3 → T
4 → F
5 → T

dfs(0)

dfs(1)

dfs(2)

dfs(3)

dfs(4)

dfs(5)

dfs(6)

dfs(7)

dfs(8)

dfs(9)

dfs(10)

dfs(11)

dfs(12)

dfs(13)

dfs(14)

dfs(15)

dfs(16)

dfs(17)

dfs(18)

dfs(19)

dfs(20)

dfs(21)

dfs(22)

dfs(23)

dfs(24)

dfs(25)

dfs(26)

dfs(27)

dfs(28)

dfs(29)

dfs(30)

dfs(31)

dfs(32)

dfs(33)

dfs(34)

dfs(35)

dfs(36)

dfs(37)

dfs(38)

dfs(39)

dfs(40)

dfs(41)

dfs(42)

dfs(43)

dfs(44)

dfs(45)

dfs(46)

dfs(47)

dfs(48)

dfs(49)

dfs(50)

dfs(51)

dfs(52)

dfs(53)

dfs(54)

dfs(55)

dfs(56)

dfs(57)

dfs(58)

dfs(59)

dfs(60)

dfs(61)

dfs(62)

dfs(63)

dfs(64)

dfs(65)

dfs(66)

dfs(67)

dfs(68)

dfs(69)

dfs(70)

dfs(71)

dfs(72)

dfs(73)

dfs(74)

dfs(75)

dfs(76)

dfs(77)

dfs(78)

dfs(79)

dfs(80)

dfs(81)

dfs(82)

dfs(83)

dfs(84)

dfs(85)

dfs(86)

dfs(87)

dfs(88)

dfs(89)

dfs(90)

dfs(91)

dfs(92)

dfs(93)

dfs(94)

dfs(95)

dfs(96)

dfs(97)

dfs(98)

dfs(99)

dfs(100)

dfs(101)

dfs(102)

dfs(103)

dfs(104)

dfs(105)

dfs(106)

dfs(107)

dfs(108)

dfs(109)

dfs(110)

dfs(111)