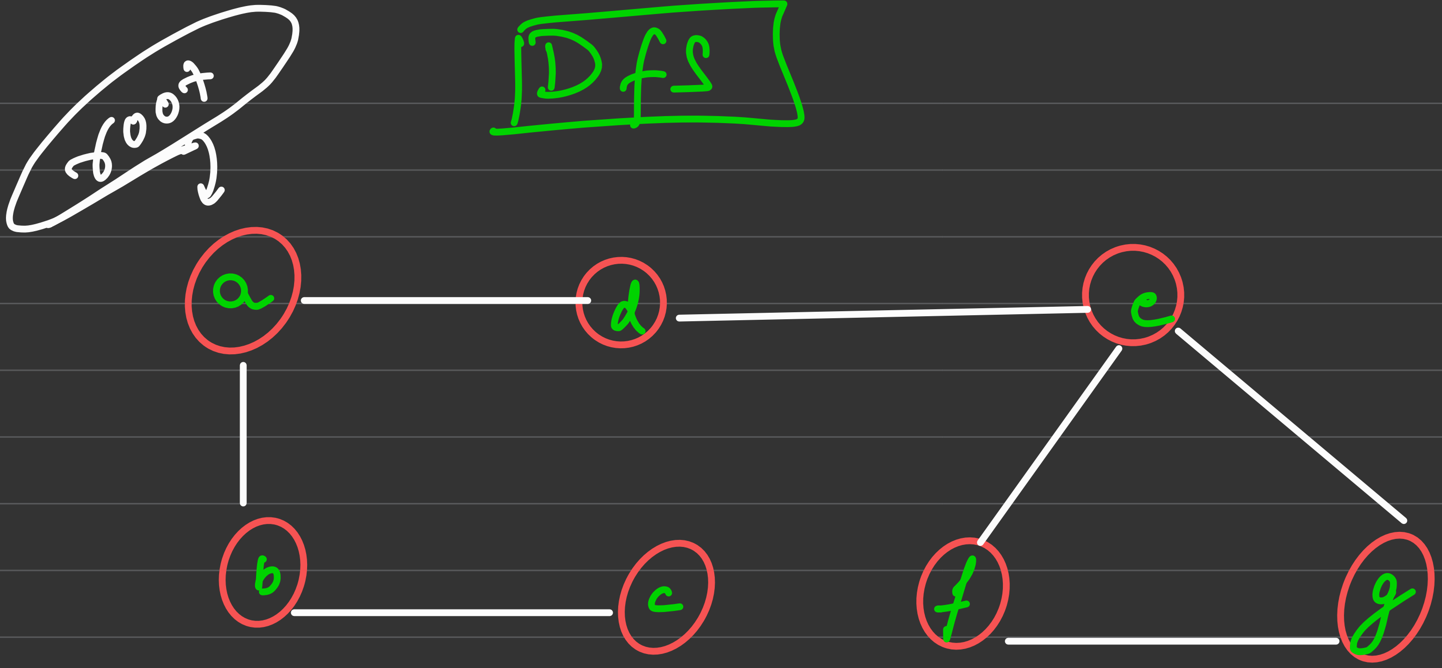
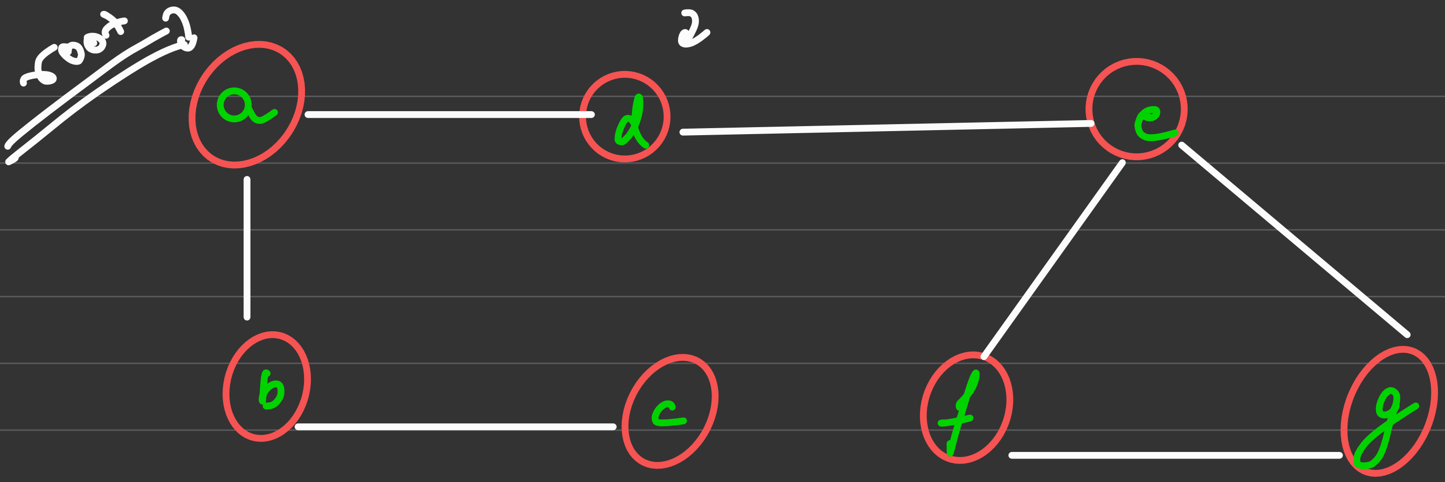
Dfs



Start at any one node and explore one branch as far as possible.



a → d → e → g → f → b → c

visited

$O(V + E)$

V → #of vertices

E → #of edges

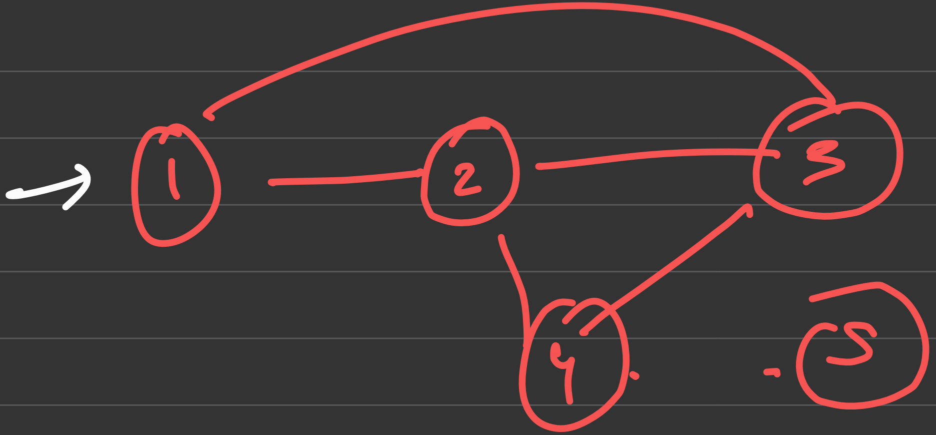
Space of BFS → Set → visited set O(v)

queue → vertices O(V)

DFS → Set → visited O(v)

rec call stack O(v)

vct



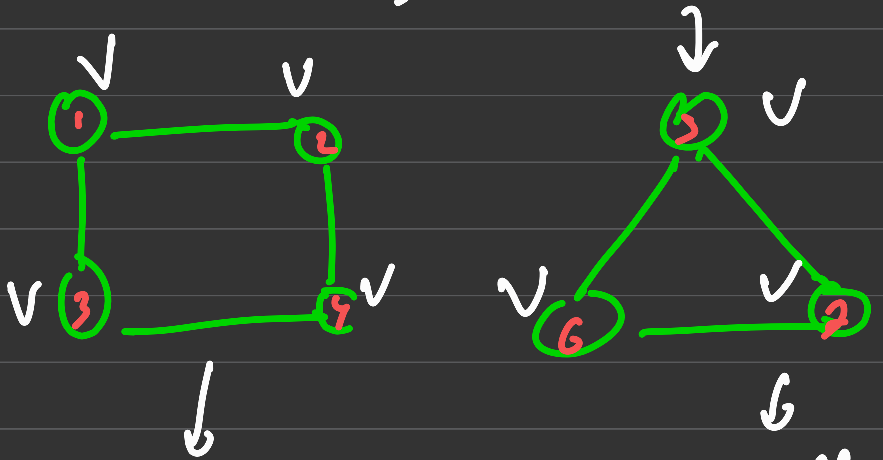
6

1 2 4 5 3

Connected Components

count the no. of connected component

CC = ~~1~~ 2



one component

another component

DFS

whenever we go inside a loop, we explore new CC.

Inside a component we have a path from one node to other. (bidirectional graph)

Q:- Given 2 integer arrays P and Q, which are some permutation of each other. You have M golden pairs (i, j). With these you can perform a swap operation on array P. i.e swap P[i] with P[j]. Tell, whether after some swap operation can you convert array P to array Q. 10^5

(2, 4)

P → 1 3 2 4

Q → 1 4 2 3

→ Yes

	1	2	3	4	5	
Ex →	1	5	4	2	3	← P
→	3	2	1	5	4	← Q

P' <1, 4, 3>

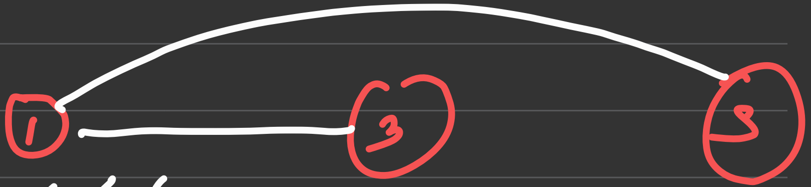
Q' <3, 1, 4>

m → (1, 3)

(1, 5)

→ (4, 2)

good pairs → indexes



~~1~~ ~~4~~ ~~3~~ 1   ~~4~~ ~~1~~ 3   ~~3~~ 4

1 3 4 → swap(1, 4)

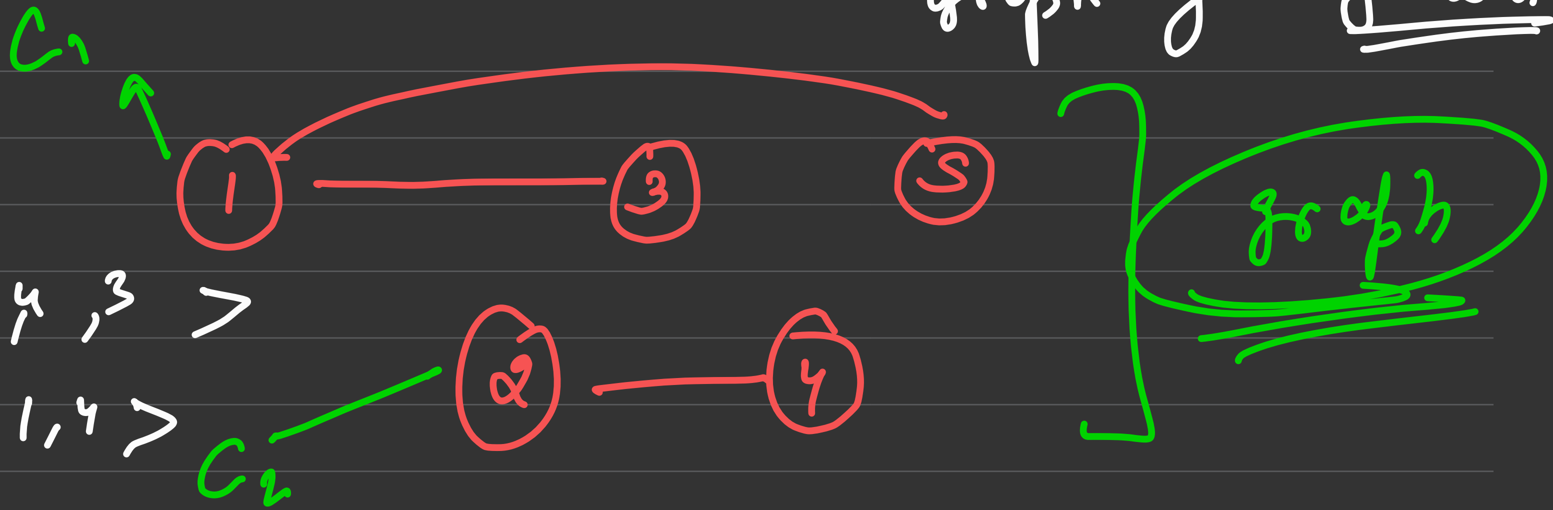
swap(4, 3)

swap(1, 3)

graph of array indices

$P' \langle 1, 4, 3 \rangle$

$Q' \langle 3, 1, 4 \rangle$



if your indexes lie in 1 connected component you can arrange them in any permutation possible.

DFS → no of golden pairs → M

$M \times 10^5$

M → edges

N size of arr

↓

vertices

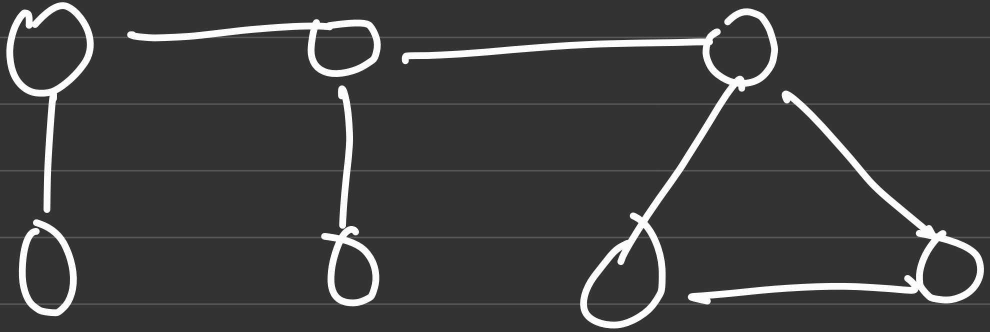
$(N+M) + N\log n$

↓

DFS

$O(N+M+N\log n)$

parent



Whether the graph has a cycle or not ? ?