Build order -) we have list of projects and
dependencies, (a,b) -) b is dependent
on a so, before b, a needs to be
completed first

(a, b, c, d, e, f)

$(a,b), (f,b), (b,d), (f,a) (d,c)$

-) f e a b d c
we can also use numbers instead of characters

$a \longrightarrow b \rightarrow d \rightarrow c$
f (pointing to a and b)

This is a problem of Topological sort

can be done using dFS or BFS (Kahn's algo)
u -) v -) u needs to be done before V

We visit each Vertex   0 -)

dfs(0) -) we cannot   1 -)
go anywhere so add it to   2 -) 3
stack
dfs(1) -) same as above   3 -) 1
dfs(2)
dfs(3)   4, 0, 1
4   5 -) 0, 2
5 -) visited so
cannot go anywhere

T	T	T			
F	F	F	F	F	F
0	1	2	3	4	5

s 4 2 3 1 0
Topological sort

Trying to run it for cyclic

$0 \rightarrow 1 \rightarrow 2$
$\downarrow 3$
4 (pointing back)

T		T		
F	F		F	F
0	1	2	3	4

0
1
3
4
2

-) So we cannot
detect here if
it possible or not

$dfs(0) \rightarrow df(1) \rightarrow dfs(2) \rightarrow \times$
$\rightarrow dfs(3) \rightarrow dfs(4) \rightarrow 0 \cdot visited$

Using Kahn's algo or BFS

$S \rightarrow 0$ (with 4 pointing to it)
$\rightarrow 1$
$\rightarrow 3$
2

Integer array -) (No. of incoming
edges)

1	0	1	0	0	0
2	2	1	1	0	0
0	1	2	3	4	5

add to queue
This becomes 0 If indegree = 0
add to queue   (no dependency)

$0 \rightarrow 3 \rightarrow$

Queue:
T	T	T	T	F	
0	1	2	3	4	5
not traversed

top order
5 4 2 0 3 1

Checking same algo for cyclic

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3$
$\downarrow 4$

Integer -) Nothing with
0 indegree

0	2	1	1	2
0	1	2	3	4

There is nothing with indegree 0

0

So, it fails if there is a
cycle ✓

$0 \rightarrow 1$
$1 \rightarrow 2$
$2 \rightarrow 3, 4$
$3 \rightarrow 4$
$4 \rightarrow 1$