

Day - 205Graph - 9

* Prerequisite Tasks:

 \Rightarrow

$$\text{Pre} = \{ \{1, 0\}, \{2, 1\}, \{3, 2\} \}$$

$$N = 4, \quad P = 3$$

 \Rightarrow

So, $\{1, 0\}$ in the pre means that Task 1 will do after completing Task 0.

 $=$

P tells the no. of Prerequisite.

 $=$

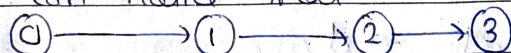
N tells the no. of tasks.

 \Rightarrow

we have to find that we can do all the tasks.

 $=$

So we can notice that —

 $=$

It something like topological sort.

 $=$

So, if the graph have cycle that means TS can't find.

 $=$

So, we can apply any cycle detection technique here.

 $=$

we can use Kahn's algo.

 $=$

First, we calculate the indegree.

 $=$

Then, push that element that have zero indegree in the queue.

* Alien Dictionary:

dict =
b a a
a b c d
a b c a
c a b
c a d

have to alien
⇒ Here, we find the order of dictionary.
⇒ We can see that, if we compare—

[b a a
a b c d

⇒ b is coming before a.

⇒ b → a

⇒ In the same way,

d → a

a → c

b → d

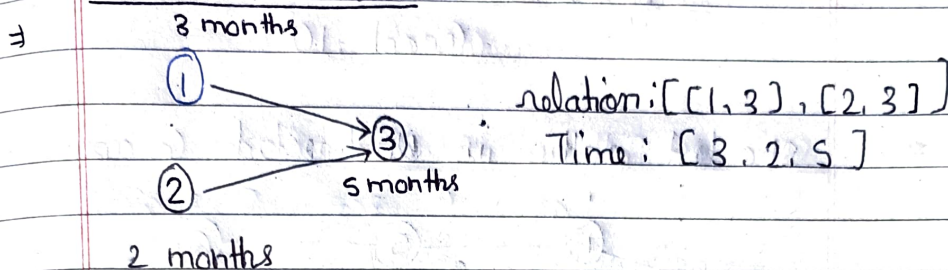
⇒ So, by using Topological Sort, we can find the order easily.

⇒ Our tasks will be —

→ Create adjacency List.

→ Apply Kahn's Algo.

* Parallel course:



\Rightarrow We have to find min. time in which we can complete all the courses.

\Rightarrow But there is a condition that we have to complete the previous course like for $[1, 3] \rightarrow$ we have to first complete course 1 then we can start course 3.

\Rightarrow Also, we can do as many course as we want at a time.

\Rightarrow So, we will start 1 & 2 then we do 3.

\Rightarrow So, we can see that this question is like Topological Sort.

\Rightarrow We will take two vector, one for indegree & one for calculating max. time.

\Rightarrow When we are decreasing indegree, we will calculate ~~write~~ the max time for that course.