

0을 완료하기 위해서는 1을 끝내야 한다는 것을 [0, 1] 쌍으로 표현하는 n개의 코스가 있다. 코스 개수 n과 이 쌍들을 입력으로 받았을 때 모든 코스가 완료 가능한지 판별하라.

Input: numCourses = 2, prerequisites = [[1,0]]

Output: true

1.dfs

import collections

class Solution:

def canFinish(self, numCourses: int, prerequisites: List[List[int]]) -> bool:

def dfs(c, p):

if finish[c] == True:

return True

if c in p[:-1]:

return False

for pr in pre[c]:

if dfs(pr, (p + [pr])[:]) == False:

return False

return True

pre = collections.defaultdict(list)

for c, p in prerequisites:

pre[c].append(p)

finish = [False if c in pre else True for c in range(numCourses)]

for i in range(len(finish)):

finish[i] = dfs(i, [i])

if finish[i] == False:

return False

return True

import collections

class Solution:

def canFinish(self, numCourses: int, prerequisites: List[List[int]]) -> bool:

def dfs(i):

if i in traced:

return False

if i in visited:

return True

traced.add(i)

for y in graph[i]:

if not dfs(y):

return False

traced.remove(i)

visited.add(i)

return True

graph = collections.defaultdict(list)

for x, y in prerequisites:

```
graph[x].append(y)
traced = set()
visited = set()
for x in list(graph):
    if not dfs(x):
        return False
return True
```