

DFS Spanning Tree란,

Jaewon Kim v

방향성은 거스르지 않게 정점들을 나열하도시

같은 위상들의 서열은 [원 → 2] 순 ([::-1]로 for문 돌릴 시 [2 → 원] 순) 으로 매개는 알고리즘이다

구현 방법:

indegree = 0 인 정점들로 DFS 함수를 호출.

visited 배열을 이용해 재방문 방지

재귀호출 더 이상 일어나지 않는 순으로 정점을 stack에 넣는다.

정점들을 위상정렬, 같은 위상은 [원 → 2] 순

visited = [True] + [False] * n

stack = []

def DFS(v):

visited[v] = True

for v2 in edge[v]:

if visited[v2] == False:

DFS(v2)

stack.append(v)

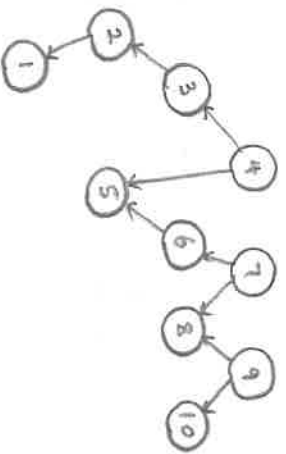
for i in range(1, n+1): # n: 정점의 개수

if indegree[i] == 0:

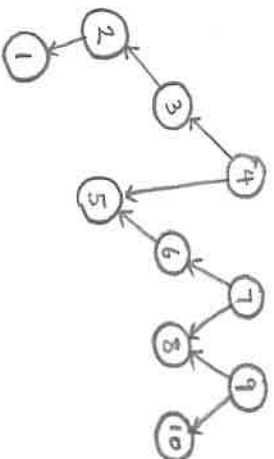
DFS(i)

print(*stack)

ex1)



ex2)



정점들을 위상정렬, 같은 위상은 [2 → 원] 순

visited = [True] + [False] * n

stack = []

def DFS(v):

visited[v] = True

for v2 in edge[v][::-1]:

if visited[v2] == False:

DFS(v2)

stack.append(v)

for i in range(n, 0, -1):

if indegree[i] == 0:

DFS(i)

print(*stack)

DFS Spanning Tree 식 위상정렬:

1 2 3 5 4 6 8 7 10 9

그래프 특징: 순마노드 → 자손노드 이하로 연결된 간선이 없다.

∴ 위상 정렬된 데로 큰 숫자를 넣으면 방향성을 거스르지 않는

가장 큰 숫자를 얻을 수 있다.

DFS Spanning Tree 식 위상정렬, 같은 위상은 [2 → 원] 순:

10 8 9 5 6 7 1 2 3 4

∴ 위상 정렬된 데로 큰 숫자를 넣으면 방향성을 거스르지 않는

가장 작은 숫자를 넣을 수 있다.

order [10] = 9

order [8] = 8

order = [, 3, 2, 1, 0, 6, 5, 4, 8, 7, 9]

order [9] = 7