pearson

# example code

|  |
| --- |
| def dfs(graph, start\_node):  from collections import deque    visit = []  stack = deque()    #시작 노드 설정해주기  stack.append(start\_node)    # 방문이 필요한 리스트가 아직 존재한다면  while stack:    # 시작 노드를 지정하고(마지막 데이터 추출(스텍 구조 활용))  node = stack.pop()    #만약 방문한 리스트에 없다면  if node not in visit:    # 방문 리스트에 노드를 추가  visit.append(node)    # 인접 노드들을 방문 예정 리스트에 추가  stack.extend(graph[node])    return visit  graph = {  'A': ['B', 'C'],  'B': ['A', 'D', 'E'],  'C': ['A', 'F', 'G'],  'D': ['B'],  'E': ['B', 'H'],  'F': ['C'],  'G': ['C'],  'H': ['C']  }  print("dfs 결과: ", dfs(graph,'A')) |

# testing result

|  |
| --- |
| **dfs 결과: ['A', 'C', 'G', 'F', 'B', 'E', 'H', 'D']** |