

Python/알고리즘

[알고리즘] SCC 찾기 - 코사라주 알고리즘(kosaraju's algorithm)

AI 꿈나무 · 2021. 4. 18. 14:19

코사라주 알고리즘(kosaraju's algorithm)

- dfs를 활용하여 그래프 내의 ssc를 찾는 알고리즘
- 시간 복잡도 $O(V+E)$
- 정방향 그래프에서 dfs를 실행하여 post가 높은 정점이 sCc의 source가 되고,
- 역방향 그래프에서 dfs를 실행하여 post가 높은 정점이 scc의 sink가 된다.
- 찾은 source와 sink를 묶어 scc를 생성한다.

작동 원리

- (1) 방향 그래프 내에서 dfs를 실행하여, dfs 탐색을 마친 정점을 stack에 삽입한다.
- (2) 역방향 그래프를 생성한다.
- (3) stack에서 정점을 추출하여, 역방향 그래프에서 dfs를 실행한다.
- (4) 방문하는 정점을 scc 에 삽입한다.
- (5) 탐색이 종료되면, scc
- (6) stack에서 방문하지 않은 정점을 추출하여, 역방향 그래프에서 dfs를 실행

파이썬 코드

```
V, E = map(int, input().split())
visited = [0] * (V+1) # visited 초기화
graph = [[] for i in range(V + 1)] # 빈 그래프 생성

# 주어진 간선에 따라 그래프 채워넣기
for _ in range(E):
    a, b = map(int, input().split())
    graph[a].append(b)

# dfs 재귀 함수
def dfs(v, visited, stack):
    visited[v] = 1

    for w in graph[v]:
        if visited[w] == 0:
            stack.append(w)
            dfs(w, visited, stack)
    stack.append(v) # 탐색을 마친 노드 stack에 저장.

# 역방향 그래프 생성
def reverseGraph():
    reverse_graph = [[] for i in range(V+1)]
```

```
for i in range(1, V+1):
    for j in graph[i]:
        reverse_graph[j].append(i)
return reverse_graph

# 역방향 그래프에 dfs 진행
def reverseDfs(v, visited,stack):
    visited[v] = 1
    stack.append(v)
    for w in reverse_graph[v]:
        if visited[w] == 0:
            reverseDfs(w, visited, stack)

# 코사라주 알고리즘
stack = []
for i in range(1, V+1):
    if visited[i] == 0:
        dfs(i, visited, stack) # stack에 탐색을 마친 정점 순으로 저장

reverse_graph = reverseGraph() # 역방향 그래프 생성

visited = [0] * (V+1) # visited 초기화
results = [] # ssc를 담을 result 생성

while stack:
    ssc = []
    node = stack.pop() # stack에서 ssc의 source 추출
    if visited[node] == 0:
        reverseDfs(node, visited, ssc) # dfs를 진행하면서, ssc의 source부
        터 탐색을 진행한다. 탐색한 정점은 ssc에 저장
        results.append(sorted(ssc)) # 재귀가 끝난 정점이 ssc의 sink
```

관련 문제

2150번: Strongly Connected Co...

첫째 줄에 두 정수 $V(1 \leq V \leq 10,000)$, $E(1 \leq E \leq 100,000)$ 가 주어진다. 이는 그래프가 V 개의 정점과 E 개의 간선으로 ...

www.acmicpc.net

출처

www.geeksforgeeks.org/strongly-connected-components/
wondy1128.tistory.com/130

♡ 1

...

구독하기

Tag

Algorithm

kosaraju

SSC

strongly connected component

알고리즘

코사라주 알고리즘

파이썬

'Python/알고리즘'의 다른글

이전글 [백준 파이썬] 2252번 줄 세우기

현재글 : [알고리즘] SCC 찾기 - 코사라주 알고리즘(kosaraju's algorithm)

관련글

- [알고리즘] 유향 그래프에서 순환(cyclic) 판별

2021.04.20
- [알고리즘] 이분 그래프(bipartite graph) 판별하기

2021.04.19
- [알고리즘] 위상정렬

2021.04.17

댓글 0

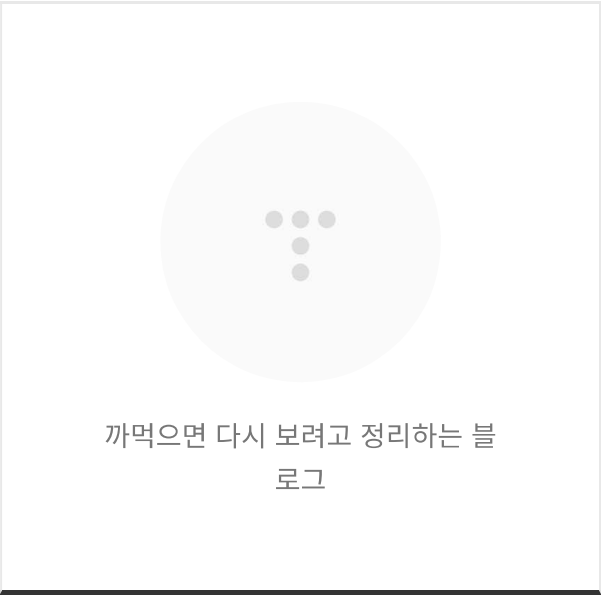
이름

비밀번호

내용을 입력해주세요.

☐ 비밀글

등록



분류 전체보기 (813)

논문 구현 (27)

논문 읽기 (254)

- Self-Supervised (42)
- Semi-Supervised (8)
- weakly (4)
- Zero shot (36)
- Meta, Few-Shot (2)
- Classification (50)
- Segmentation (19)
- Object Detection (40)
- Video Recognition (17)
- Video Segmentation (0)
- Augmentation (3)
- 3D Vision (4)
- Active Learning (1)

- GAN (5)
- NLP (7)
- Style Transfer (1)
- Face Recognition (1)
- Image Processing (3)
- Pose Estimation (1)
- long tail (2)
- Model Compression (1)
- Optimization (2)
- Activation Function (1)
- Regularization (1)
- Normalization (3)

Python (257)

- PyTorch 공부 (62)
- 파이썬 OpenCV 공부 (88)
- 알고리즘 (52)
- 백준 (30)
- 기타 코딩 (25)

수학 (251)

- 딥러닝 이론 (43)
- 해석학 (1)
- Multiple View Geometry (5)
- Statistical Learning (27)
- Bayesian (3)
- 정보이론 (3)
- 벡터 미적분학 (20)
- 선형대수학 (36)
- 확률론 (21)
- 기초 통계학 (81)
- cs224w (3)
- 기타 정리 (8)

기타 (16)

- 영어 공부 (16)
- 보호글 (0)

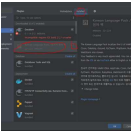
최근글 인기글

[Python] list()와 []의 차이점

2022.06.16 01:55

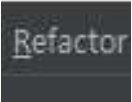
Selenium을 사용해서 백준 유저가 푼 문제 크롤링 하기

2022.06.15 23:35



파이참 한국어에서 영어로 설정하기

2022.06.12 17:52



파이참 버전 업데이트

2022.06.12 17:49

[Pytorch] List를 Tensor로 변경하기. torch.stack

2022.05.22 23:14

검색내용을 입력하세요.



딥러닝 공부방 구독하기



Copyright © Kakao Corp. All rights reserved.

관련사이트

+