

강의 14: 그래프 II: 깊이 우선 탐색

강의 개요

- 깊이 우선 탐색
- 간선 분류
- 순환 검출
- 위상 정렬

기억해야 할 점:

- 그래프 탐색: 그래프를 탐색하는 것
예시: 시작점 s 에서 원하는 정점까지 가는 경로를 찾는 것
- 인접 리스트: $|V|$ 연결 리스트의 배열 Adj
 - 각 정점 $u \in V$ 에 대해, $\text{Adj}[u]$ 는 u 의 이웃들을 저장한다, 예: $\{v \in V \mid (u,v) \in E\}$ (방향 그래프에서 (u,v) 는 나가는 간선이다)

예시:

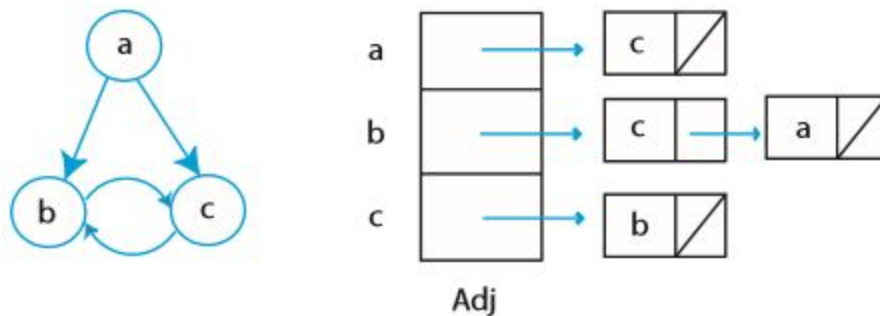


그림 1: 인접 리스트

너비 우선 탐색 (BFS):

s 에서 레벨별로 그래프를 탐색한다 — 최단 경로를 찾는다

깊이 우선 탐색 (DFS)

미로를 탐험하는 것과 같다.

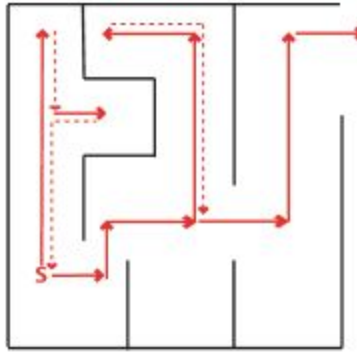


그림 2: 깊이 우선 탐색의 Frontier

깊이 우선 탐색 알고리즘

- 막다른 골목까지 경로를 따라간다
- 탐색되지 않은 이웃에 도달할 때까지 뺑 부스러기를 따라 역추적한다
- 재귀적으로 탐색한다
- 정점을 반복하지 않도록 조심한다

```

parent = {s: None}

DFS-visit (v, Adj, s):
    start → for v in Adj[s]:
        v      if v not in parent:
                parent[v] = s
                DFS-visit (v, Adj, v)
    finish → v

DFS(V, Adj)
    parent = {}
    for s in V:
        if s not in parent:
            parent[s] = None
            DFS-visit (v, Adj, s)
  
```

search from start vertex s
(only see stuff reachable from s)

explore entire graph
(could do same to extend BFS)

그림 3: 깊이 우선 탐색 알고리즘

예시

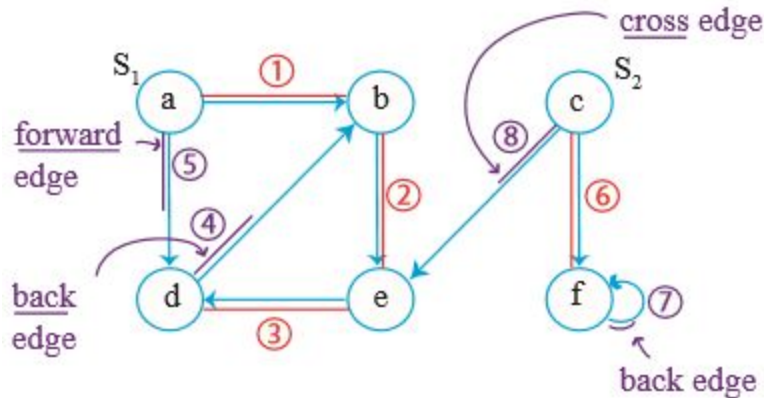


그림 4: 깊이 우선 탐색

간선 분류

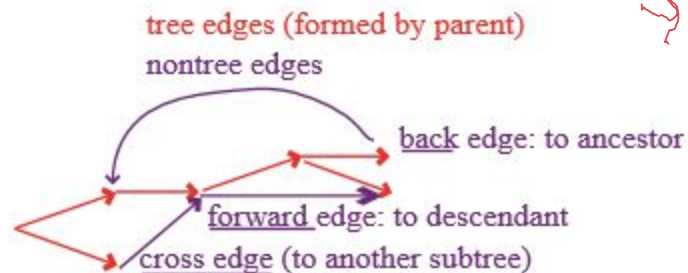


그림 5: 간선 분류

① 트리간선
 → ② 순방향
 ③ 역방향
 ④ 교차간선
 ↳ 선조관계

- 이 분류를 계산하기 위해 (역방향 또는 방향), 정점을 “스택 상에” 있는 동안 **이웃 간선** 표시한다
- 무방향 그래프에서는 트리 간선과 역방향 간선만 있다

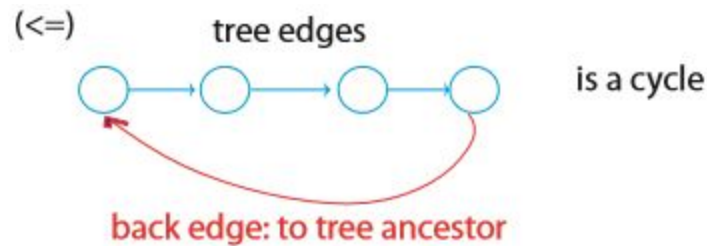
분석

- 깊이 우선 탐색 방문은 DFS-visit gets called with a vertex s only once (parent[s] 가 설정되기 때문에)
 \Rightarrow 깊이 우선 탐색 방문 시간 = $\sum_{s \in V} |Adj[s]| = O(E)$
- 깊이 우선 탐색 외부 루프는 $O(V)$ 를 더한다
 $\Rightarrow O(V + E)$ 시간 (**선형 시간**)

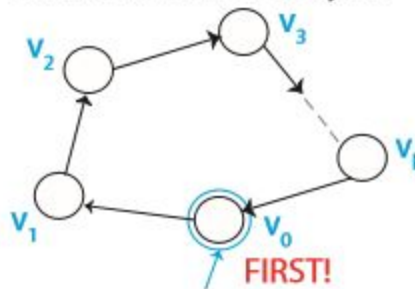
순환 검출

그래프 G 는 순환이 있다 \Leftrightarrow 깊이 우선 탐색은 역방향 간선이 있다

증명



(\Rightarrow) consider first visit to cycle:



- v_i 방문이 끝나기 전,
 v_{i+1} 을 방문하고 끝낼 것이다:
 (v_i, v_{i+1}) 간선을 고려한다
 $\Rightarrow v_{i+1}$ 을 지금 방문하거나 이미 방문했을 것이다
- $\Rightarrow v_0$ 방문이 끝나기 전,
 v_k 를 방문할 것이다(& 전에 방문한 적이 없다)
- $\Rightarrow v_k$ (또는 v_0) 방문이 끝나기 전,
 (v_k, v_0) 간선을 역방향 간선으로 볼 것이다

잡 스케줄링

주어진 비순환성 방향 그래프 (DAG)에서, 정점은 할 일이고 간선은 의존 상태를 의미할 때, 의존 상태를 위반하지 않고 할 일을 정렬한다

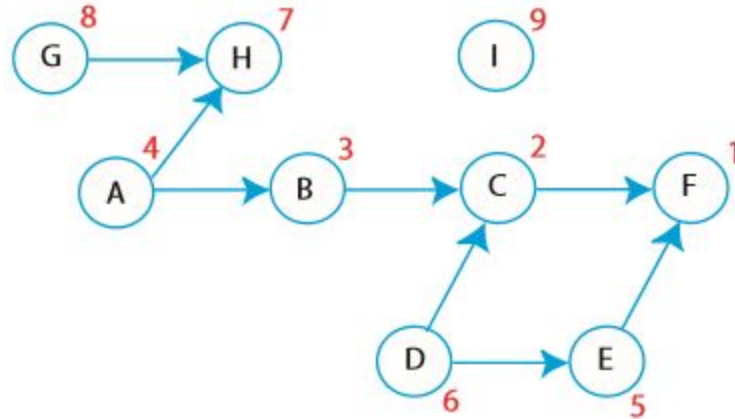


그림 6: 의존 그래프: 깊이 우선 탐색 마무리 시간

소스:

소스 = 들어오는 간선이 없는 정점

= 처음 (A,G,I)때 스케줄을 짤 수 있다

시도:

각 소스에서 너비 우선 탐색:

- A에서 A, BH, C, F를 찾는다
- D에서 D, BE, CF를 찾는다 ← **느리고.. 틀렸다!**
- G에서 G, H를 찾는다
- I에서 I를 찾는다

위상 정렬

깊이 우선 탐색 **마무리 시간**의 역 (깊이 우선 탐색(v)가 끝나는 시간)

```

DFS-Visit(v)
...
order.append(v)
order.reverse()
  
```

정확성

어떤 간선 (u,v) 에 대해서 — u 는 v 이전에 정렬된다, 예: u 전에 v 가 끝난다



- 만약 v 전에 u 가 방문되면:
 - u 방문이 끝나기 전, v 를 방문할 것이다 ((u,v) 를 통하거나 다른 방법을 통해)
 - $\Rightarrow u$ 전에 v 가 끝난다
- 만약 u 전에 v 가 방문되면:
 - 그래프는 비순환적이다
 - $\Rightarrow v$ 에서 u 로 도달할 수 없다
 - $\Rightarrow u$ 를 방문하기 전에 v 방문이 끝난다

MIT OpenCourseWare

<http://ocw.mit.edu>

6.006 Introduction to Algorithms

Fall 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.