1 그래프의 순회-깊이 우선 탐색, 너비 우선 탐색

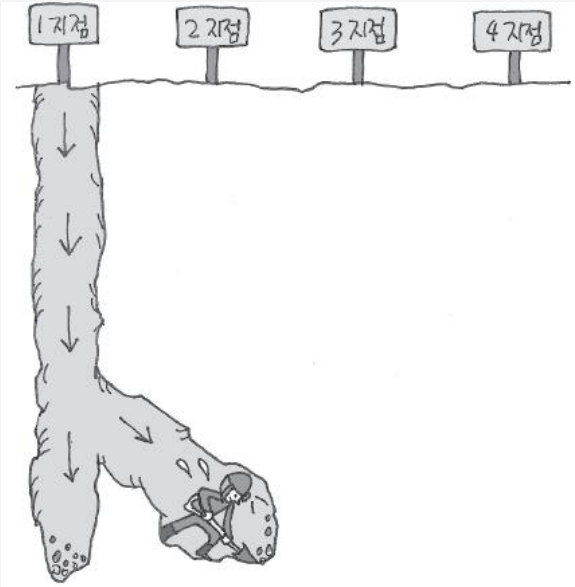
1 그래프 순회(Graph traversal), 그래프 탐색(Graph search)

- 하나의 정점에서 시작하여 그래프에 있는 모든 정점을 한번씩 방문하여 처리하는 연산

- 그래프 탐색방법
  - 깊이 우선 탐색(Depth first search : DFS)
  - 너비 우선 탐색(Breadth first search : BFS)

2 그래프 순회의 예 : 우물 파기

- 한 지점을 골라서 팔 수 있을 때까지 계속해서 깊게 파다가 아무리 땅을 파도 물이 나오지 않으면, 밖으로 나와 다른 지점을 골라서 다시 깊게 땅을 파는 방법 (→깊이 우선 탐색)



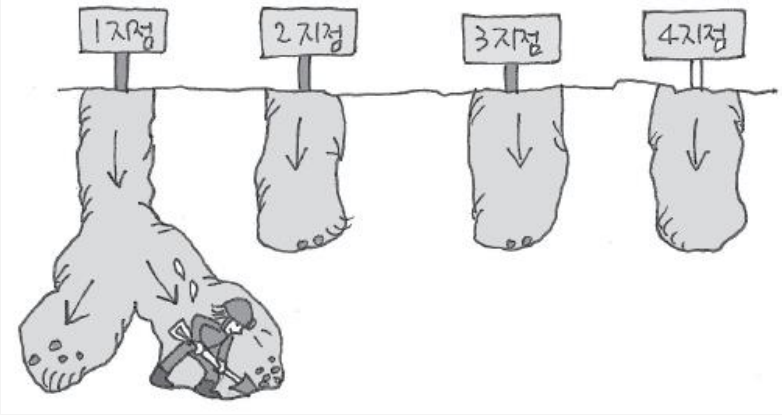
(a) 깊이 우선 탐색 예

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 그래프 순회의 예 : 우물 파기

- 여러 지점을 고르게 파보고 물이 나오지 않으면, 파놓은 구덩이들을 다시 좀더 깊게 파는 방법 (→ 너비 우선 탐색)

(b) 너비 우선 탐색 예



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

- 순회 방법
  - 시작 정점의 한 방향으로 갈 수 있는 경로가 있는 곳까지 깊이 탐색해 가다가 더 이상 갈 곳이 없으면, 가장 마지막에 만났던 갈림길 간선이 있는 정점으로 되돌아와 다른 방향의 간선으로 탐색을 계속 반복하여 결국 모든 정점을 방문하는 순회방법
  - 가장 마지막에 만났던 갈림길 간선의 정점으로 가장 먼저 되돌아가서 다시 깊이 우선 탐색을 반복해야 하므로 후입선출 구조의 스택 사용

3 깊이 우선 탐색

▶ 깊이 우선 탐색 순서

❶ 시작 정점 v를 결정하여 방문한다.

❷ 정점 v에 인접한 정점 중에서

❷-ⓐ 방문하지 않은 정점 w가 있으면 정점 v를 스택에 push하고 w를 방문한다. 그리고 w를 v로 하여 다시 ❷를 반복한다.

❷-ⓑ 방문하지 않은 정점이 없으면 스택을 pop하여 받은 가장 마지막에 방문한 정점을 v로 설정한 뒤 다시 ❷를 수행한다.

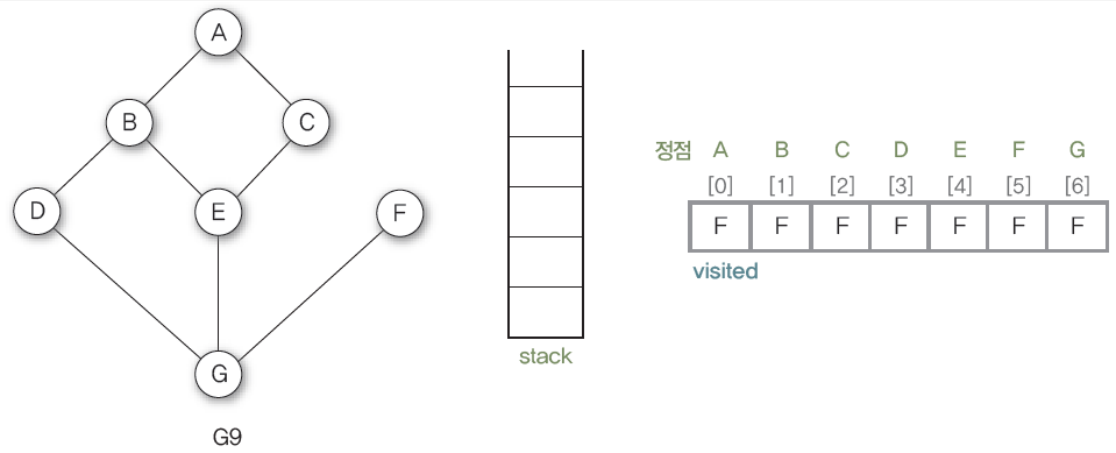
❸ 스택이 공백이 될 때까지 ❷를 반복한다.

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

▶ 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

① 그래프 G9의 깊이 우선 탐색을 위한 초기 상태 : 배열 visited를 false로 초기화하고 공백 스택 생성



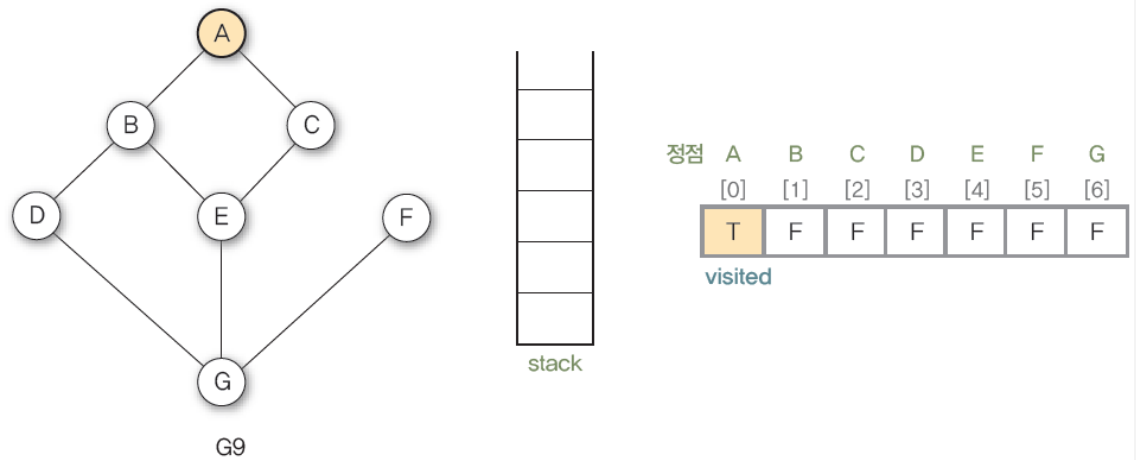
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

▶ 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

② 정점 A를 시작으로 깊이 우선 탐색을 시작

```
visited[A] ← true;
A 방문;
```



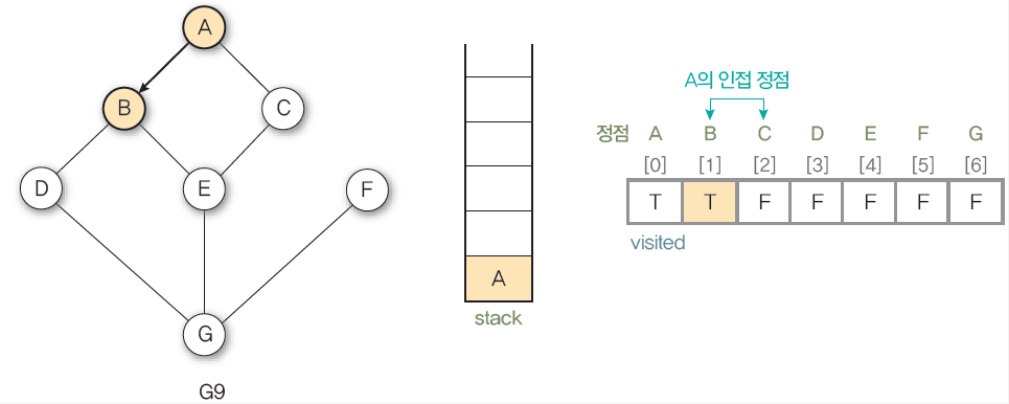
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

▶ 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

③ 정점 A에 방문하지 않은 정점 B, C가 있으므로 A를 스택에 push하고, 인접 정점 B와 C 중에서 오름차순에 따라 B를 선택하여 탐색을 계속함

```
push(stack, A);
visited[B] ← true;
B 방문;'
```



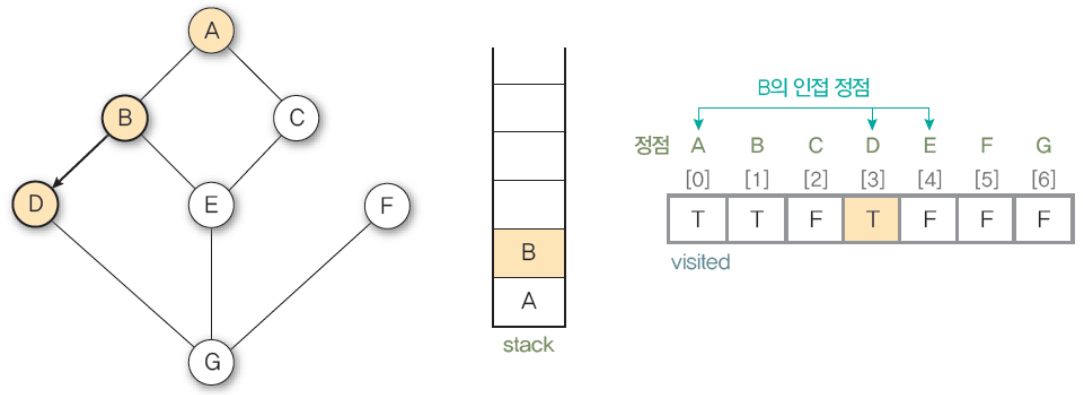
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

▶ 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

④ 정점 B에 방문하지 않은 정점 D, E가 있으므로 B를 스택에 push하고, 방문하지 않은 인접 정점 D와 E 중에서 오름차순에 따라 D를 선택하여 탐색을 계속

```
push(stack, B);
visited[D] ← true;
D 방문;
```



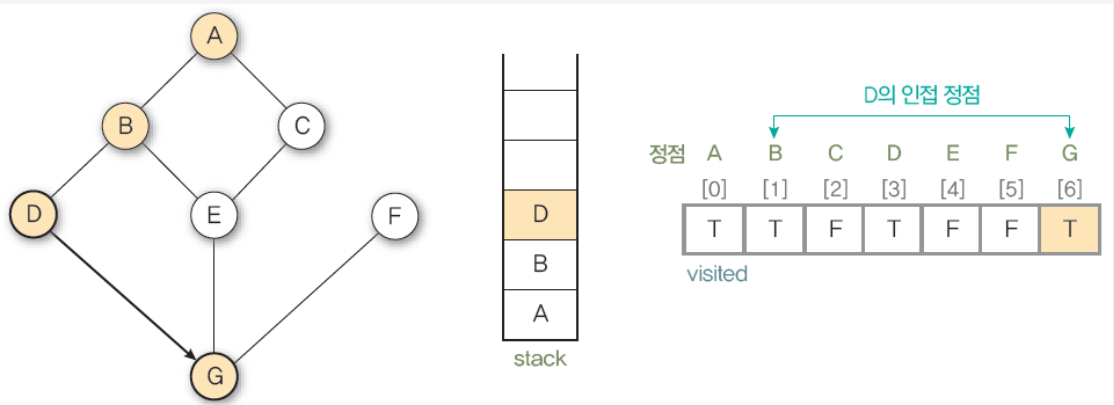
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

- 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

⑤ 정점 D에 방문하지 않은 정점 G가 있으므로 D를 스택에 push하고, 인접 정점 G를 선택하여 탐색을 계속

```
push(stack, D);
visited[G] ← true;
G 방문;
```



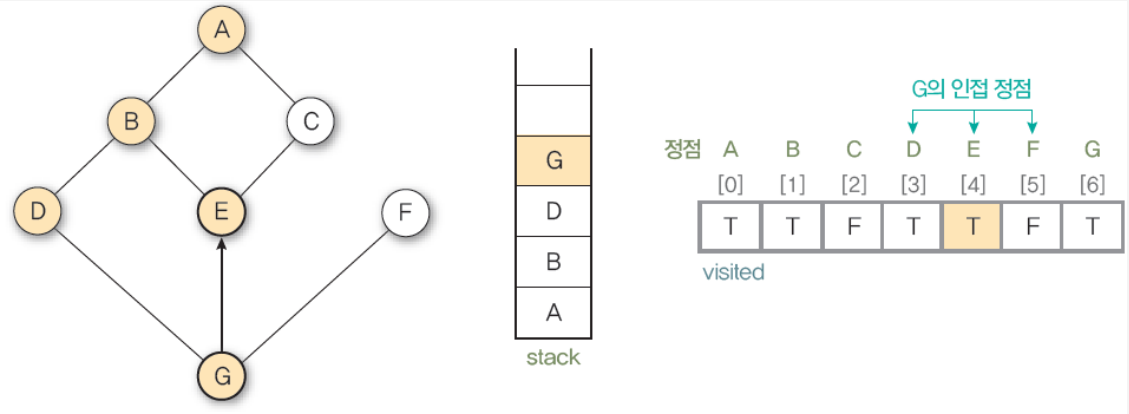
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

▶ 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

⑥ 정점 G에 방문하지 않은 정점 E, F가 있으므로 G를 스택에 push하고, 방문하지 않은 인접 정점 E와 F중에서 오름차순에 따라 E를 선택하여 탐색을 계속

```
push(stack, G);
visited[E] ← true;
E 방문;
```



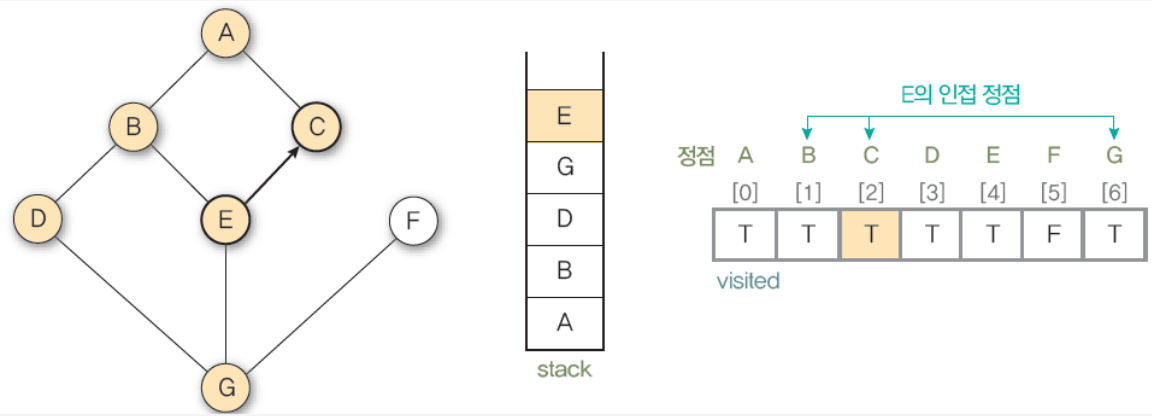
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

▶ 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

⑦ 정점 E에 방문하지 않은 정점 C가 있으므로 E를 스택에 push하고, 방문하지 않은 인접 정점 C를 선택하여 탐색을 계속

```
push(stack, E);
visited[C] ← true;
C 방문;
```



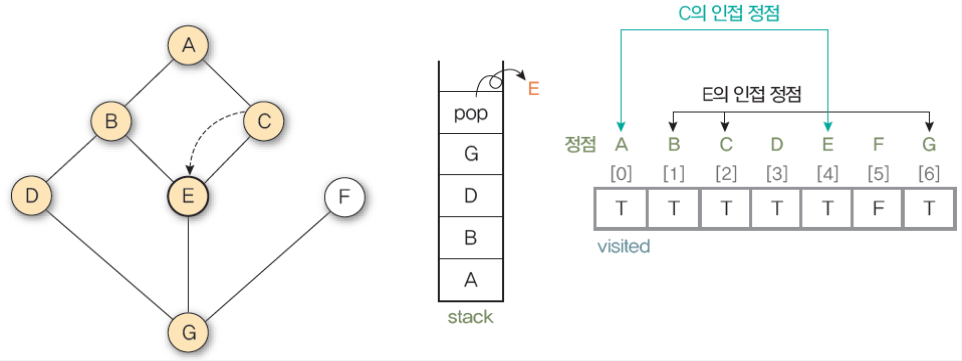
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

- 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

⑧ 정점 C에서 방문하지 않은 인접 정점이 없으므로 마지막 정점으로 돌아가기 위해 스택을 pop하여 받은 정점 E에 대해서 방문하지 않은 인접 정점이 있는지 확인, 정점 E는 방문하지 않은 인접 정점이 없음

```
pop(stack);
```



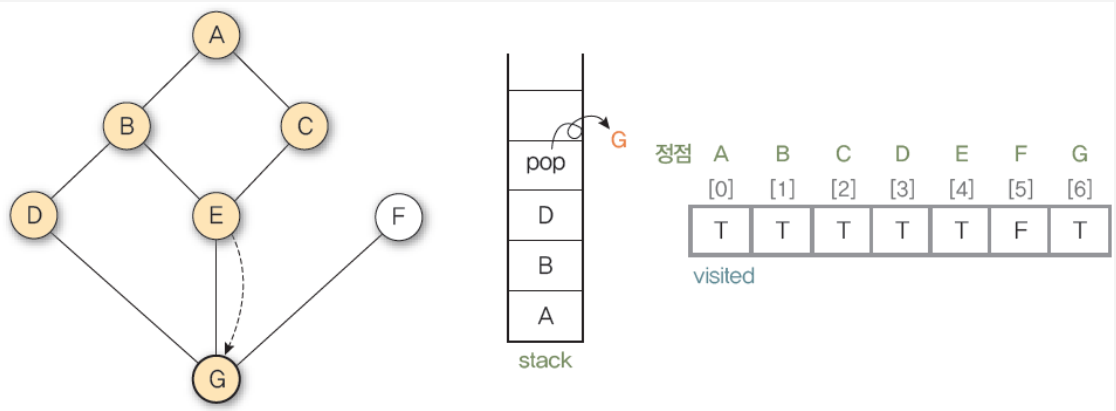
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

- 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

⑨ 현재 정점 E에서 방문할 수 있는 인접 정점이 없으므로, 다시 스택을 pop하여 받은 정점 G에 대해서 방문하지 않은 인접 정점이 있는지 확인

```
pop(stack);
```



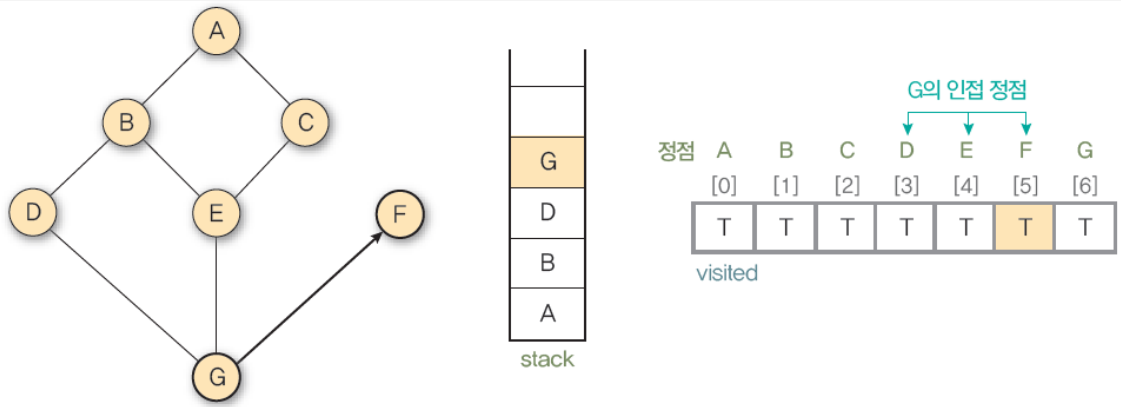
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

- 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

⑩ 현재 정점 G에서 방문하지 않은 정점 F가 있으므로 G를 스택에 push하고, 인접정점 F를 선택하여 탐색을 계속

```
push(stack, G);
visited[F] ← true;
F 방문;
```



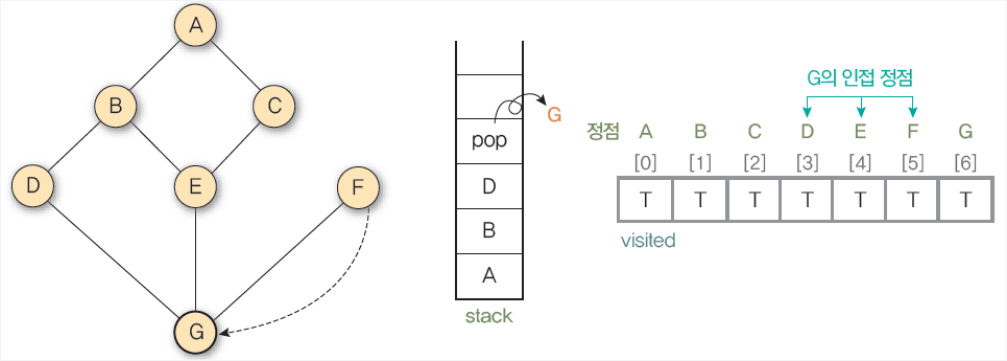
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

- 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

⑪ 현재 정점 F에서 방문하지 않은 인접 정점이 없으므로 마지막 정점으로 돌아가기 위해 스택을 pop하여 받은 정점 G에 대해서 방문하지 않은 인접 정점이 있는지 확인, 정점 G는 방문하지 않은 인접 정점이 없음

```
pop(stack);
```



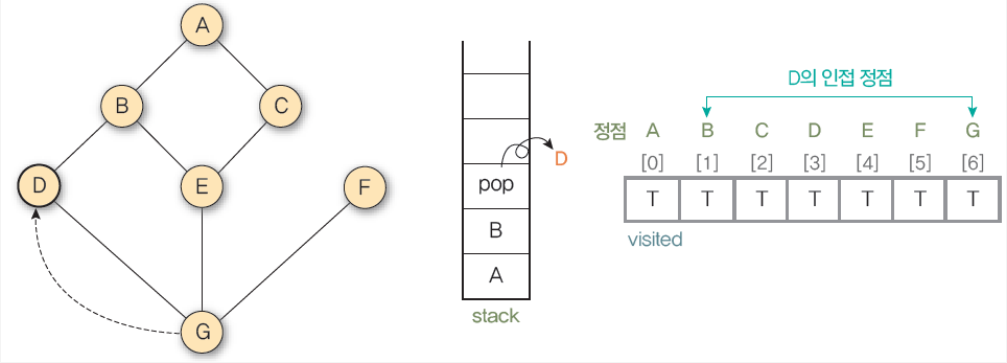
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

- 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

⑫ 현재 정점 G에서 방문하지 않은 인접 정점이 없으므로 다시 마지막 정점으로 돌아가기 위해 스택을 pop하여 받은 정점 D에 대해서 방문하지 않은 인접 정점이 있는지 확인, 정점 D는 방문하지 않은 인접 정점이 없음

```
pop(stack);
```



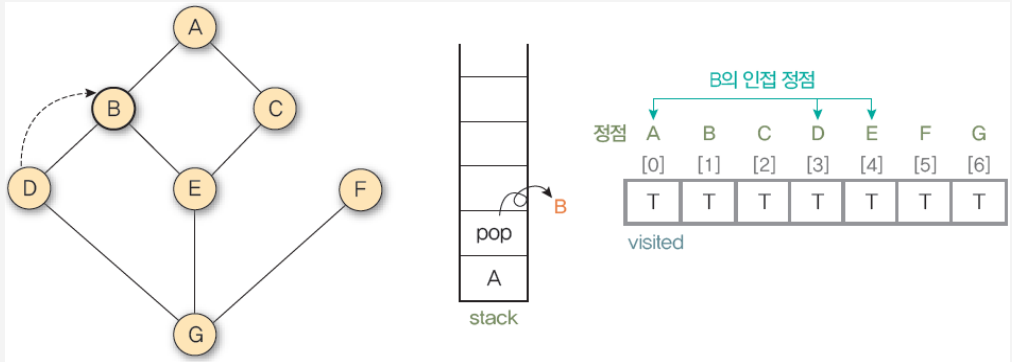
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

- 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

⑬ 현재 정점 D에서 방문하지 않은 인접 정점이 없으므로 다시 마지막 정점으로 돌아가기 위해 스택을 pop하여 받은 정점 B에 대해서 방문하지 않은 인접 정점이 있는지 확인, 정점 B는 방문하지 않은 인접 정점이 없음

```
pop(stack);
```



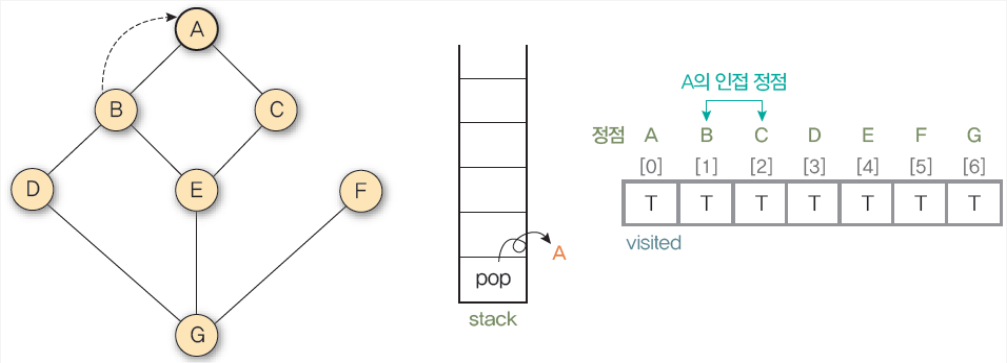
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

- 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

⑭ 현재 정점 B에서 방문하지 않은 인접 정점이 없으므로 다시 마지막 정점으로 돌아가기 위해 스택을 pop, 그리고 받은 정점 A에 대해서 방문하지 않은 인접 정점이 있는지 확인, 정점 A는 방문 하지 않은 인접 정점이 없음

```
pop(stack);
```



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

▶ 알고리즘에 따라 그래프 G9를 깊이 우선 탐색하는 과정

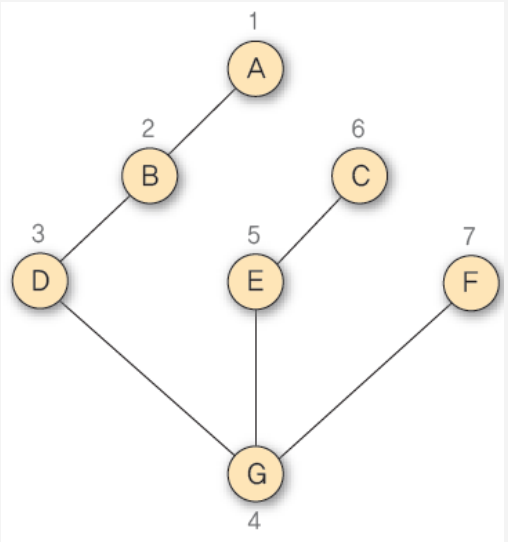
⑮ 현재 정점 A에서 방문하지 않은 인접 정점이 없으므로 마지막 정점으로 돌아가기 위해 스택을 pop, 스택이 공백이므로 깊이 우선 탐색을 종료

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 깊이 우선 탐색

▶ 그래프 G9 깊이 우선 탐색 결과

[그래프 G9의 깊이 우선 탐색 경로
: A-B-D-G-E-C-F]



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

- 시작정점에서 인접한 정점을 모두 차례로 방문하고 나서 방문했던 정점에서부터 다시 인접한 정점을 차례로 방문하는 방식

- 가까운 정점들을 먼저 방문하고 멀리 있는 정점들은 나중에 방문하는 순회방법 → 큐를 사용

4 너비 우선 탐색(Breadth first search : BFS)

▶ 너비 우선 탐색 순서

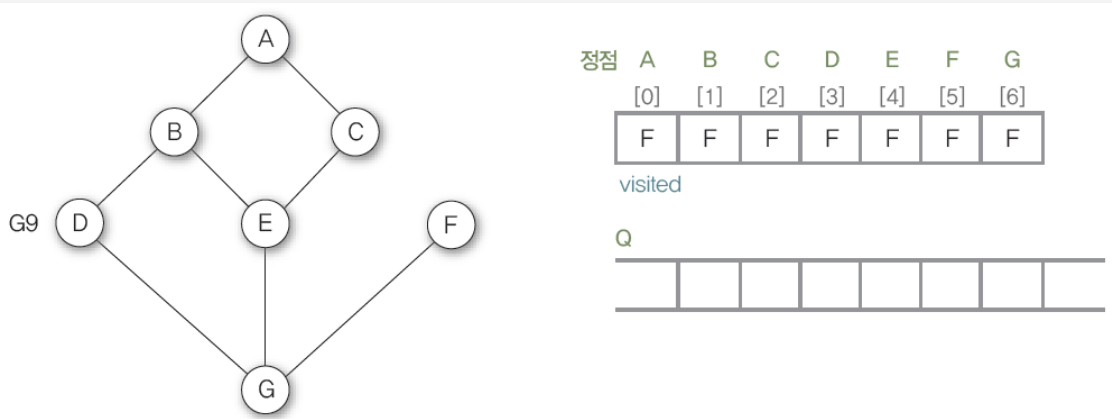
1. 시작 정점 v를 결정하여 방문한다.
2. 정점 v에 인접한 정점 중에서 방문하지 않은 정점을 차례로 방문하면서 큐에 enQueue한다.
3. 방문하지 않은 인접한 정점이 없으면, 방문했던 정점에서 인접한 정점을 다시 차례로 방문하기 위해 큐에서 deQueue 하여 받은 정점을 v로 설정하고 ❷를 반복한다.
4. 큐가 공백이 될 때까지 ❷~❸을 반복한다.

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

▶ 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

⓪ 초기상태 : 배열 visited를 False로 초기화, 공백 큐를 생성



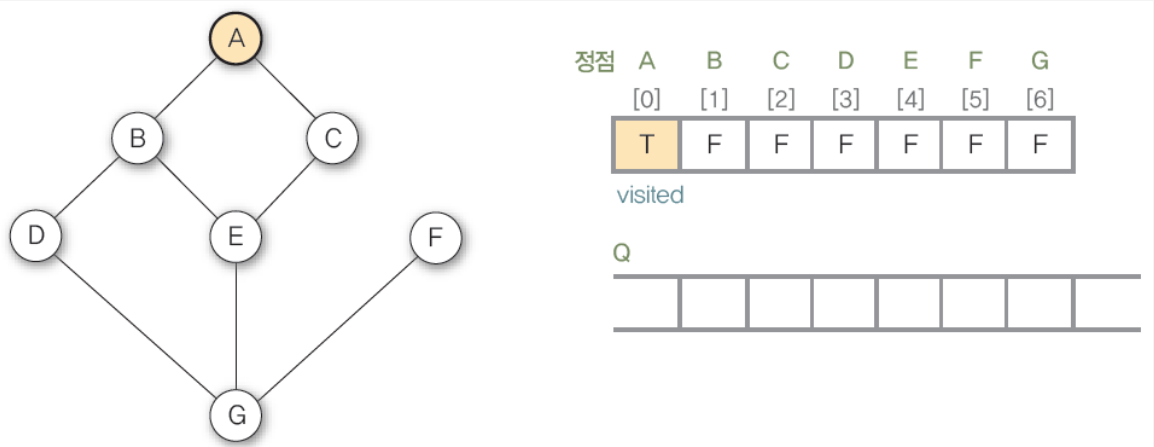
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

▶ 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

① 정점 A를 시작으로 너비 우선 탐색을 시작

```
visited[A] ← true;
A 방문;
```



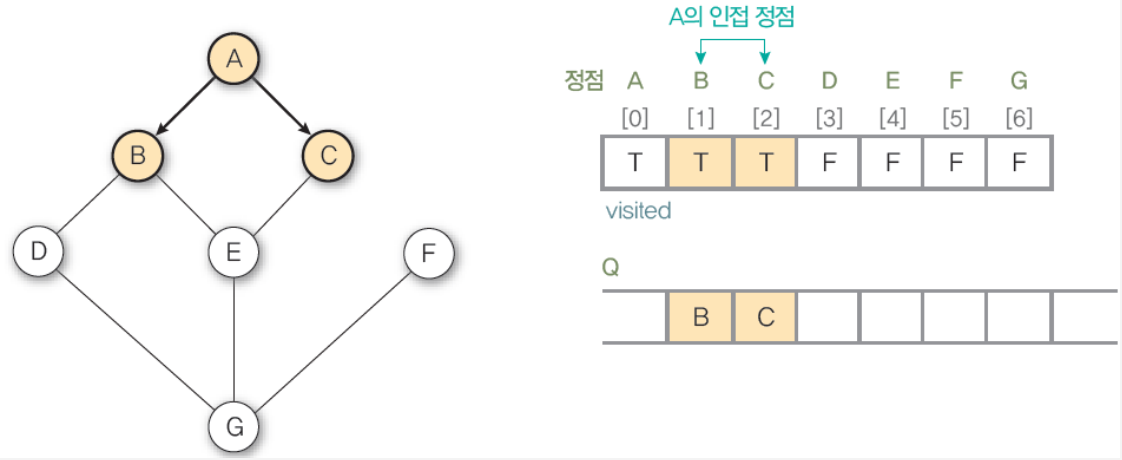
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

▶ 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

② 정점 A에서 방문하지 않은 모든 인접 정점 B, C를 방문하고 큐에 enQueue

```
visited[(A가 방문하지 않은 인접 정점 B와 C)] ← true;
(A가 방문하지 않은 인접 정점 B와 C) 방문;
enQueue(Q, (A가 방문하지 않은 인접 정점 B와 C));
```



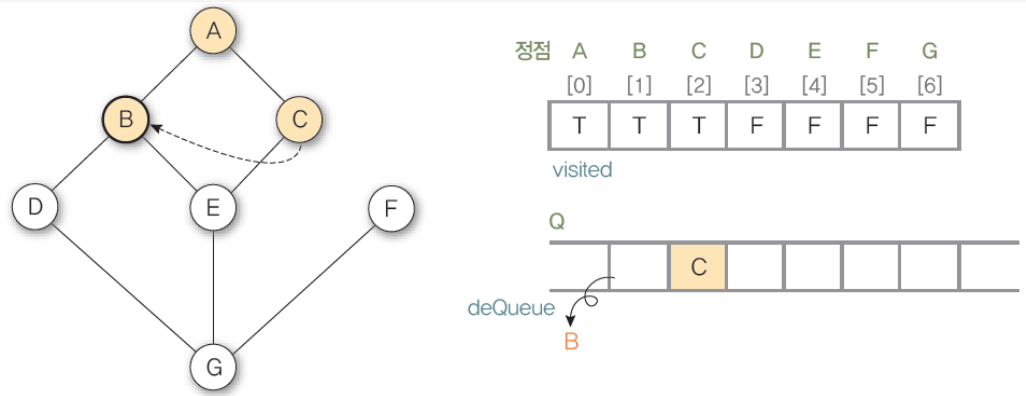
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

▶ 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

③ 정점 A에 대한 인접 정점들을 처리했으므로, 너비 우선 탐색을 계속할 다음 정점을 찾기 위해 큐를 deQueue하여 B를 받음

```
v ← deQueue(Q);
```



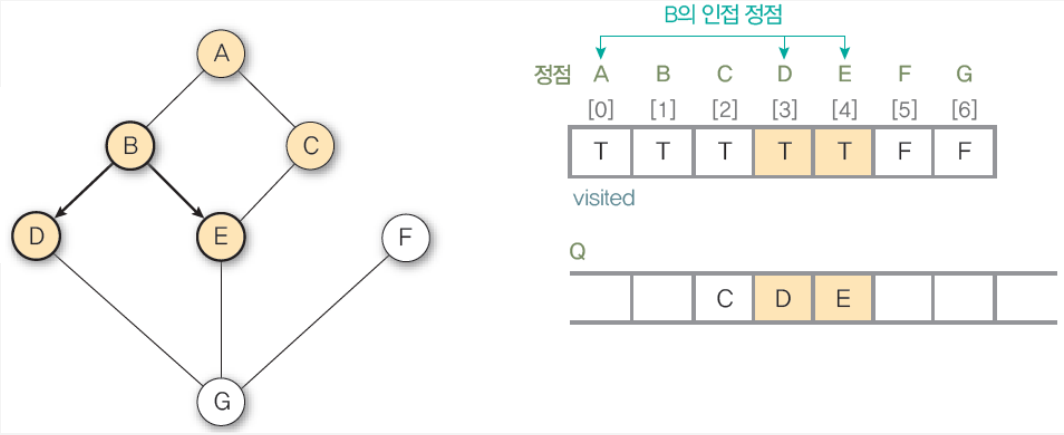
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

▶ 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

④ 정점 B에서 방문하지 않은 모든 인접 정점 D, E를 방문하고 큐에 enQueue

```
visited[(B가 방문하지 않은 인접 정점 D와 E)] ← true;
(B가 방문하지 않은 인접 정점 D와 E) 방문;
enQueue(Q, (B가 방문하지 않은 인접 정점 D와 E));
```



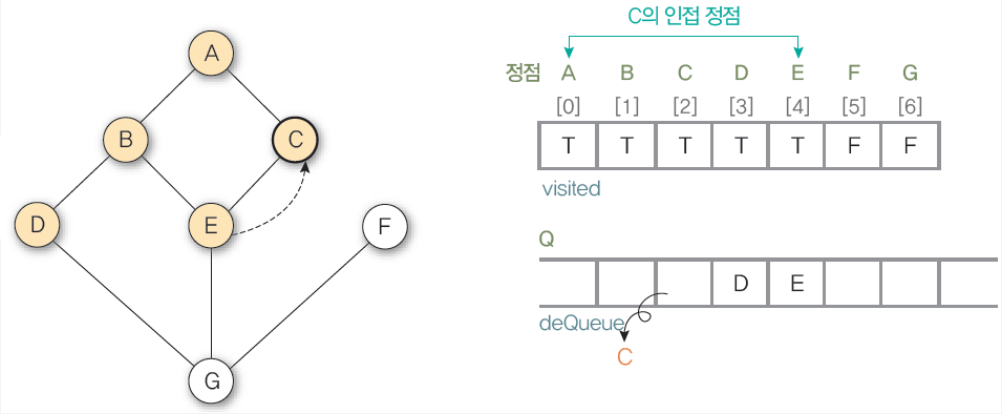
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

▶ 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

⑤ 정점 B에 대한 인접 정점들을 처리했으므로, 너비 우선 탐색을 계속할 다음 정점을 찾기 위해 큐를 deQueue하여 C를 받음

```
v ← deQueue(Q);
```



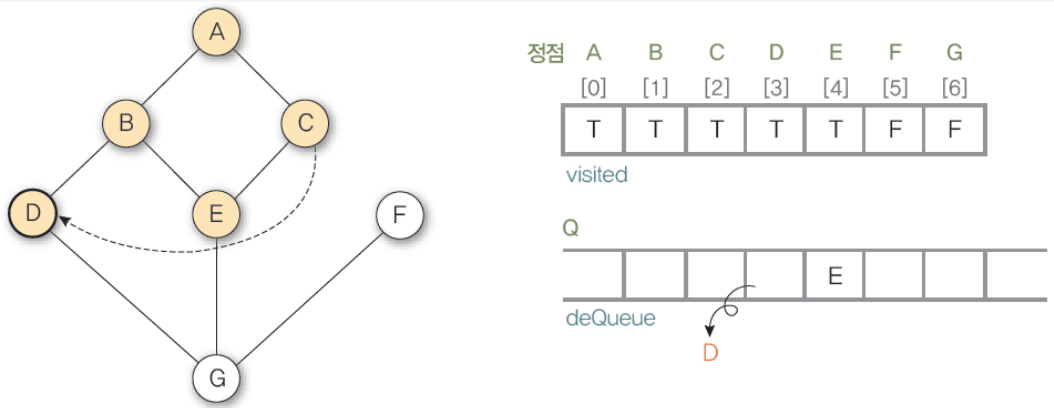
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

▶ 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

⑥ 정점 C에는 방문하지 않은 인접 정점이 없으므로, 너비 우선 탐색을 계속할 다음 정점을 찾기 위해 큐를 deQueue하여 D를 받음

```
v ← deQueue(Q);
```



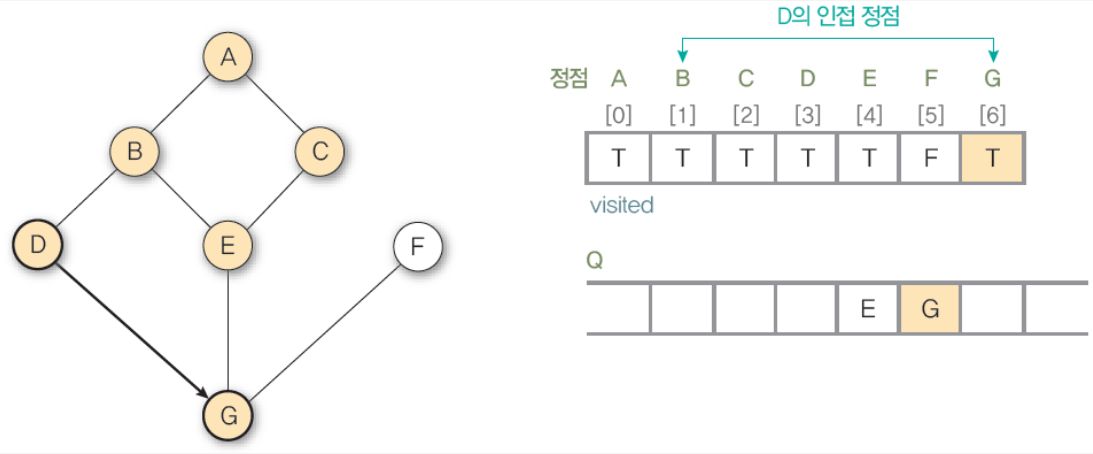
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

▶ 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

⑦ 정점 D에서 방문하지 않은 인접 정점 G를 방문하고 큐에 enQueue

```
visited[(D가 방문하지 않은 인접 정점 G)] ← true;
(D가 방문하지 않은 인접 정점 G) 방문;
enQueue(Q, (D가 방문하지 않은 인접 정점 G));
```



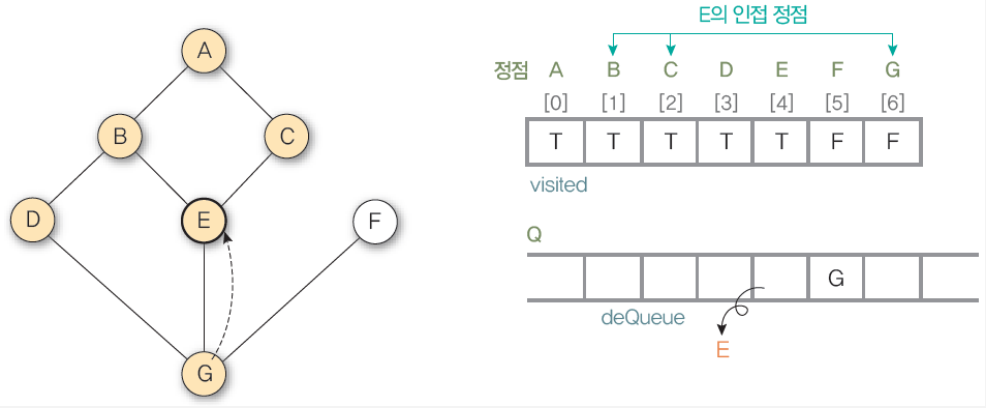
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

▶ 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

⑧ 정점 D에 대한 인접 정점들을 처리했으므로, 너비 우선 탐색을 계속할 다음 정점을 찾기 위해 큐를 deQueue하여 E를 받음

```
v ← deQueue(Q);
```



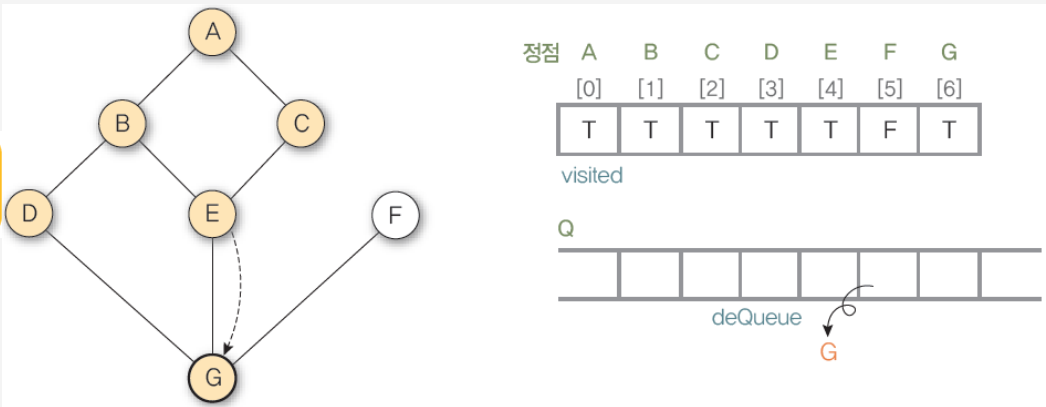
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

▶ 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

⑨ 정점 E에는 방문하지 않은 인접 정점이 없으므로, 너비 우선 탐색을 계속할 다음 정점을 찾기 위해 큐를 deQueue하여 G를 받음

```
v ← deQueue(Q);
```



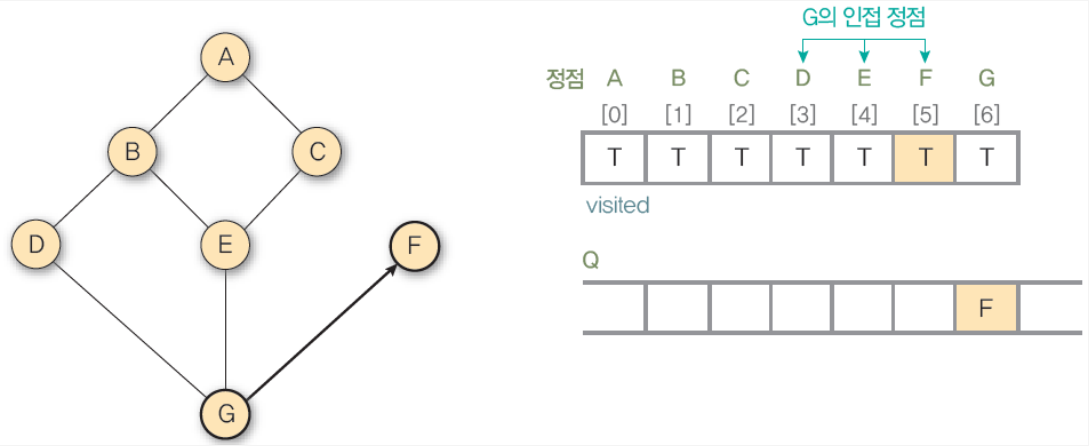
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

▶ 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

⑩ 정점 G에서 방문하지 않은 인접 정점 F를 방문하고 큐에 enQueue

```
visited[(G가 방문하지 않은 인접 정점 F)] ← true;
(G가 방문하지 않은 인접 정점 F) 방문;
enQueue(Q, (G가 방문하지 않은 인접 정점 F));
```



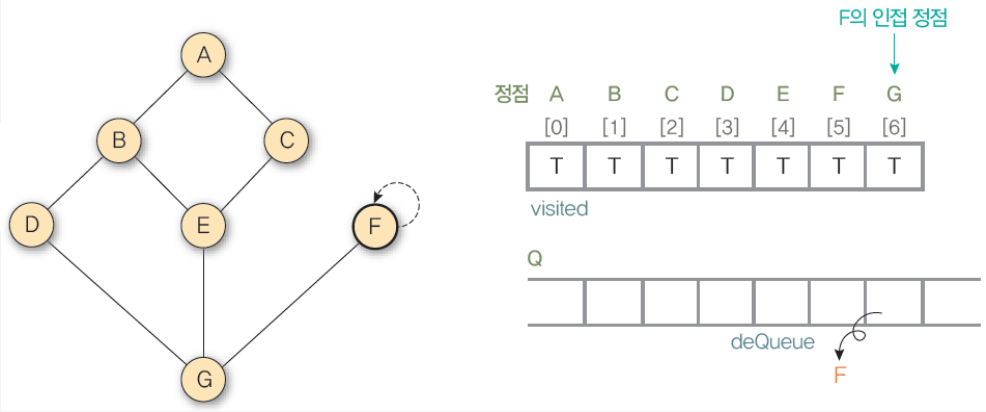
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

▶ 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

⑪ 정점 G에 대한 인접 정점들을 처리했으므로, 너비 우선 탐색을 계속할 다음 정점을 찾기 위해 큐를 deQueue하여 F를 받음

```
v ← deQueue(Q);
```



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 너비 우선 탐색(Breadth first search : BFS)

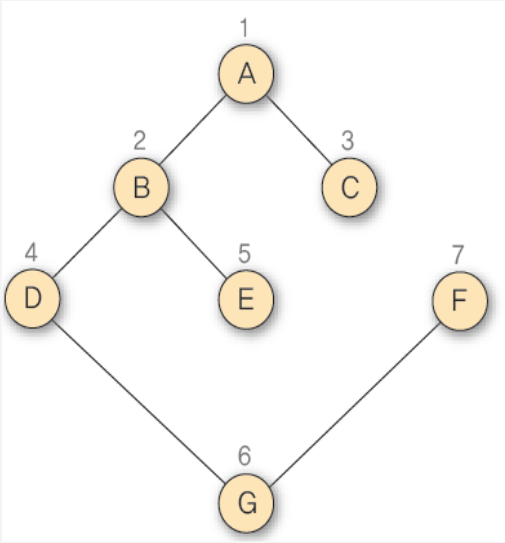
- 알고리즘에 따른 그래프 G9의 너비 우선 탐색 과정

⑫ 정점 F는 모든 인접 정점을 방문했으므로,
너비 우선 탐색을 계속할 다음 정점을 찾기 위해
큐를 deQueue, 큐가 공백이므로 너비 우선 탐색을
종료

4 너비 우선 탐색(Breadth first search : BFS)

▶ 그래프 G9 너비 우선 탐색 결과

[그래프 G9의 너비 우선 탐색 경로
: A-B-C-D-E-G-F]

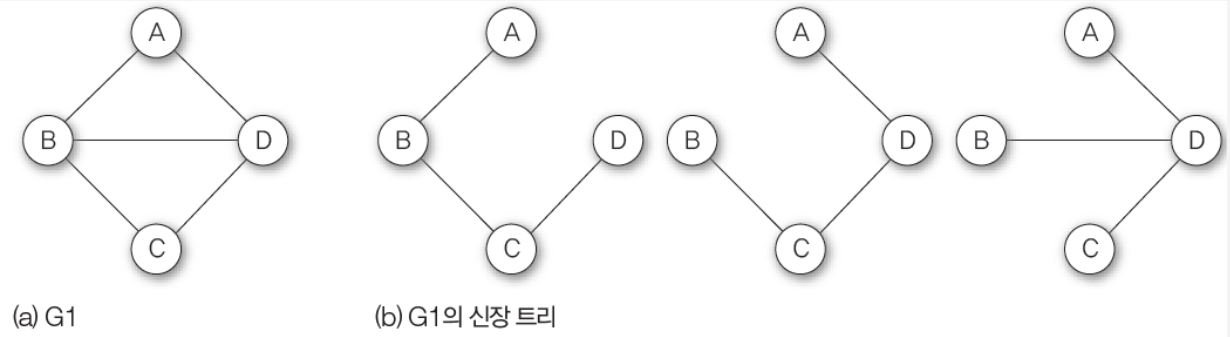


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 신장 트리

1 신장 트리(Spanning tree)

- n개의 정점으로 이루어진 무방향 그래프 G에서 n개의 모든 정점과 n-1개의 간선으로 사이클 없는 단순한 연결 그래프



[그래프 G1과 신장 트리의 예]

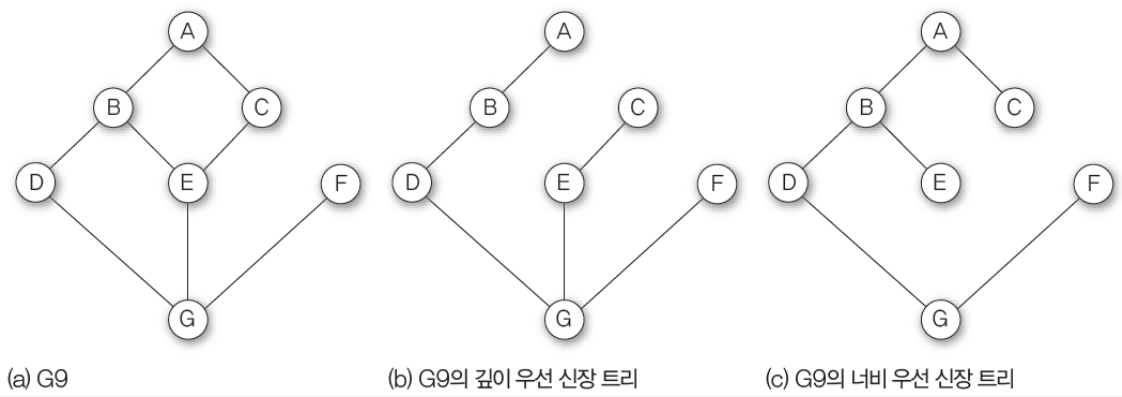
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 신장 트리(Spanning tree)

- 깊이 우선 신장 트리(Depth first spanning tree)
  - 깊이 우선 탐색을 이용하여 생성된 신장 트리
- 너비 우선 신장 트리(Breadth first spanning tree)
  - 너비 우선 탐색을 이용하여 생성된 신장 트리

1 신장 트리(Spanning tree)

[그래프 G9와 신장 트리의 예]



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 최소 비용 신장 트리

1 최소 비용 신장 트리(Minimum cost spanning tree)

- 무방향 가중치 그래프에서 신장 트리를 구성하는 간선들의 가중치 합이 최소인 신장 트리
  - 가중치
    : 그래프의 간선에 주어진 가중치비용이나 거리, 시간을 의미하는 값

- 최소 비용 신장 트리를 만드는 알고리즘
  - 크루스칼(Kruskal)의 알고리즘
  - 프림(Prime)의 알고리즘

2 크루스칼 알고리즘 I

- 가중치가 높은 간선을 제거하면서 최소 비용 신장 트리를 만드는 방법
- 크루스칼 알고리즘 I 의 순서

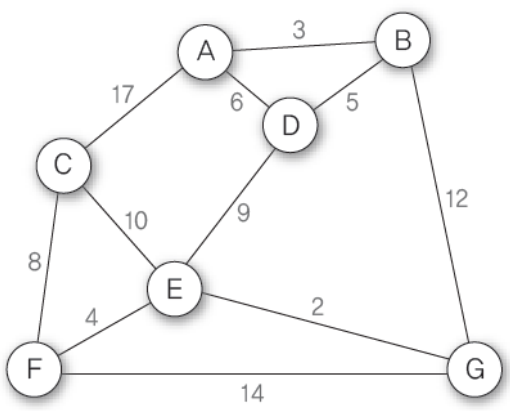
1. 그래프 G의 모든 간선을 가중치에 따라 내림차순으로 정렬한다.
2. 그래프 G에서 가중치가 가장 높은 간선을 제거한다. 단, 이때 정점을 그래프에서 분리시키는 간선은 제거할 수 없으므로 이런 경우에는 그 다음으로 가중치가 높은 간선을 제거한다.
3. 그래프 G에 간선이 n–1개만 남을 때까지 ❷를 반복한다.
4. 그래프에 간선이 n–1개만 남으면 최소 비용 신장 트리가 완성된다.

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 크루스칼 알고리즘 I

▶ 크루스칼 알고리즘 I 을 이용하여 G10의 최소 비용 신장 트리 만들기

⓪ 초기 상태
: 그래프 G10의 간선을 가중치에 따라서 내림차순 정렬



G10

가중치	간선
17	(A, C)
14	(F, G)
12	(B, G)
10	(C, E)
9	(D, E)
8	(C, F)
6	(A, D)
5	(B, D)
4	(E, F)
3	(A, B)
2	(E, G)

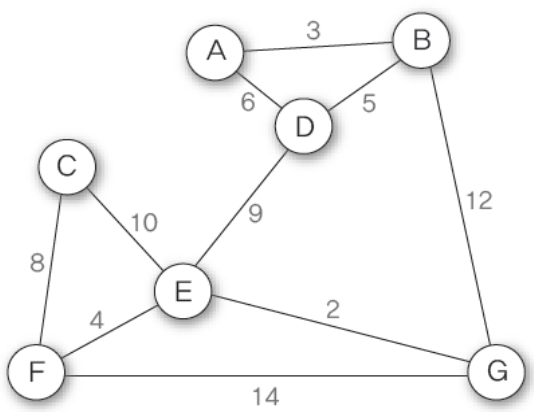
간선 수 : 11개

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 크루스칼 알고리즘 I

▶ 크루스칼 알고리즘 I 을 이용하여 G10의 최소 비용 신장 트리 만들기

① 정점 A를 시작으로
너비 우선 탐색을 시작



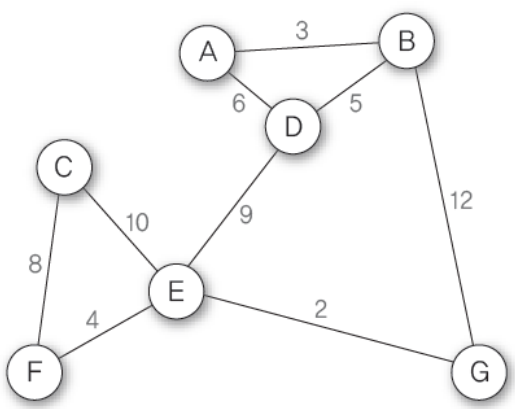
가중치	간선
~~17~~	~~(A, C)~~
14	(F, G)
12	(B, G)
10	(C, E)
9	(D, E)
8	(C, F)
6	(A, D)
5	(B, D)
4	(E, F)
3	(A, B)
2	(E, G)

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 크루스칼 알고리즘 I

▶ 크루스칼 알고리즘 I 을 이용하여 G10의 최소 비용 신장 트리 만들기

② 남은 간선 중에서 가중치가 가장 높은 간선 (F, G)를 제거
→ 남은 간선 수는 아홉 개



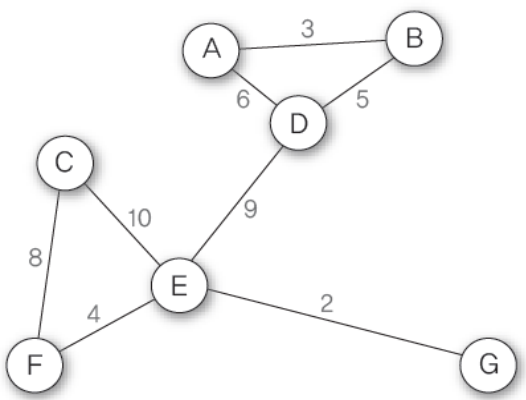
가중치	간선
~~17~~	~~(A, C)~~
~~14~~	~~(F, G)~~
12	(B, G)
10	(C, E)
9	(D, E)
8	(C, F)
6	(A, D)
5	(B, D)
4	(E, F)
3	(A, B)
2	(E, G)

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 크루스칼 알고리즘 I

▶ 크루스칼 알고리즘 I 을 이용하여 G10의 최소 비용 신장 트리 만들기

③ 남은 간선 중에서 가중치가 가장 높은 간선 (B, G)를 제거
→ 남은 간선 수는 여덟 개



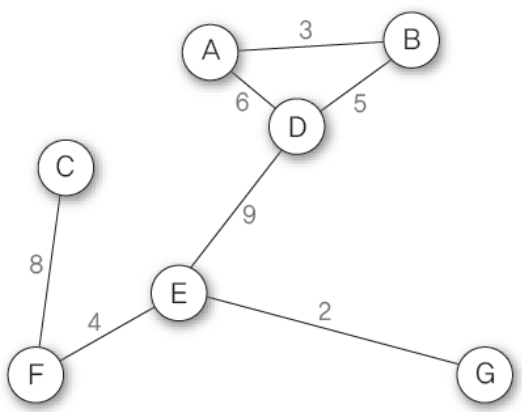
가중치	간선
~~17~~	~~(A, C)~~
~~14~~	~~(F, G)~~
~~12~~	~~(B, G)~~
10	(C, E)
9	(D, E)
8	(C, F)
6	(A, D)
5	(B, D)
4	(E, F)
3	(A, B)
2	(E, G)

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 크루스칼 알고리즘 I

▶ 크루스칼 알고리즘 I 을 이용하여 G10의 최소 비용 신장 트리 만들기

④ 남은 간선 중에서 가중치가 가장 높은 간선 (C, E)를 제거
→ 남은 간선 수는 일곱 개



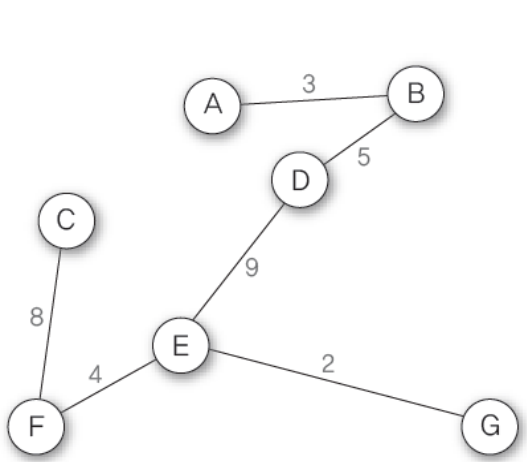
가중치	간선
~~17~~	~~(A, C)~~
~~14~~	~~(F, G)~~
~~12~~	~~(B, G)~~
~~10~~	~~(C, E)~~
9	(D, E)
8	(C, F)
6	(A, D)
5	(B, D)
4	(E, F)
3	(A, B)
2	(E, G)

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 크루스칼 알고리즘 I

▶ 크루스칼 알고리즘 I 을 이용하여 G10의 최소 비용 신장 트리 만들기

⑤ 남은 간선 중에서 가중치가 가장 높은 간선 (D, E)를 제거하면 그래프가 분리되어 단절 그래프가 되므로 그 다음으로 가중치가 높은 간선 (C, F)를 제거



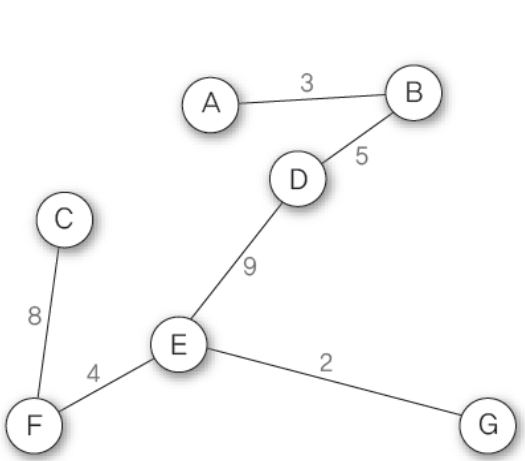
가중치	간선
~~17~~	~~(A, C)~~
~~14~~	~~(F, G)~~
~~12~~	~~(B, G)~~
~~10~~	~~(C, E)~~
9	(D, E)
8	(C, F)
~~6~~	~~(A, D)~~
5	(B, D)
4	(E, F)
3	(A, B)
2	(E, G)

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 크루스칼 알고리즘 I

▶ 크루스칼 알고리즘 I 을 이용하여 G10의 최소 비용 신장 트리 만들기

⑤ 그런데 간선 (C, F)를 제거하면 정점 C가 그래프에서 분리되므로 제거 불가능, 따라서 그 다음으로 가중치가 높은 간선 (A, D)를 제거
→ 남은 간선 수는 여섯 개



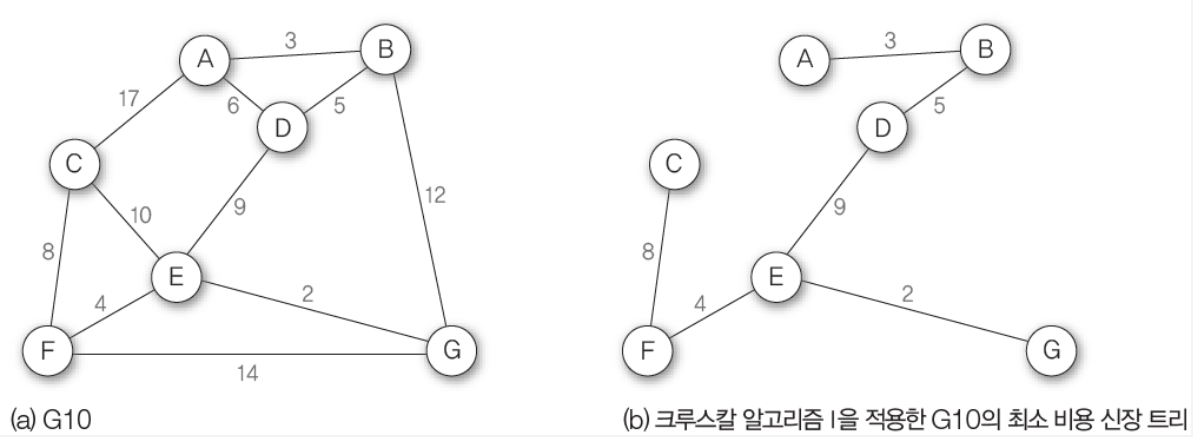
가중치	간선
~~17~~	~~(A, C)~~
~~14~~	~~(F, G)~~
~~12~~	~~(B, G)~~
~~10~~	~~(C, E)~~
9	(D, E)
8	(C, F)
~~6~~	~~(A, D)~~
5	(B, D)
4	(E, F)
3	(A, B)
2	(E, G)

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 크루스칼 알고리즘 I

- 크루스칼 알고리즘 I 을 이용하여 G10의 최소 비용 신장 트리 만들기

⑥ 현재 남은 간선 수가 여섯 개이므로 알고리즘 수행을 종료하면 신장 트리 완성



[그래프 G10을 크루스칼 알고리즘 I 으로 완성한 최소 비용 신장 트리]

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 크루스칼 알고리즘 II

- 가중치가 낮은 간선을 삽입하면서 최소 비용 신장 트리를 만드는 방법
- 크루스칼 알고리즘 II 의 순서

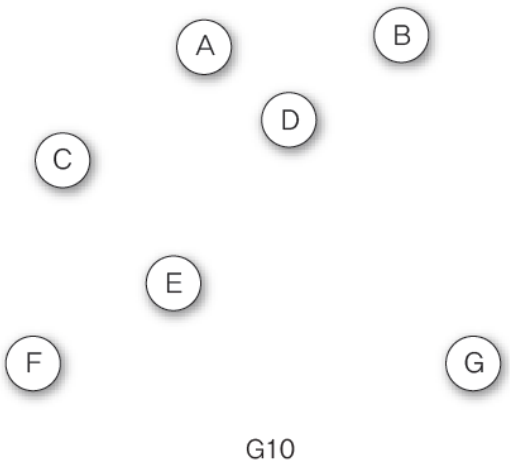
1. 그래프 G의 모든 간선을 가중치에 따라 오름차순으로 정리한다.
2. 그래프 G에 가중치가 가장 낮은 간선을 삽입한다. 단, 이때 사이클을 형성하는 간선은 삽입할 수 없으므로 그 다음으로 가중치가 낮은 간선을 삽입한다.
3. 그래프 G에 간선이 n–1개 삽입될 때까지 ❷를 반복한다.
4. 그래프 G의 간선이 n–1개가 되면 최소 비용 신장 트리가 완성된다.

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 크루스칼 알고리즘 II

▶ 크루스칼 알고리즘 II 를 이용하여 G10의 최소 비용 신장 트리 만들기

⓪ 초기 상태
: 그래프 G10의 간선 가중치에 따라 오름차순으로 정렬되어 있고, 그래프 G10은 정점만 있음



G10

가중치	간선
2	(E, G)
3	(A, B)
4	(E, F)
5	(B, D)
6	(A, D)
8	(C, F)
9	(D, E)
10	(C, E)
12	(B, G)
14	(F, G)
17	(A, C)

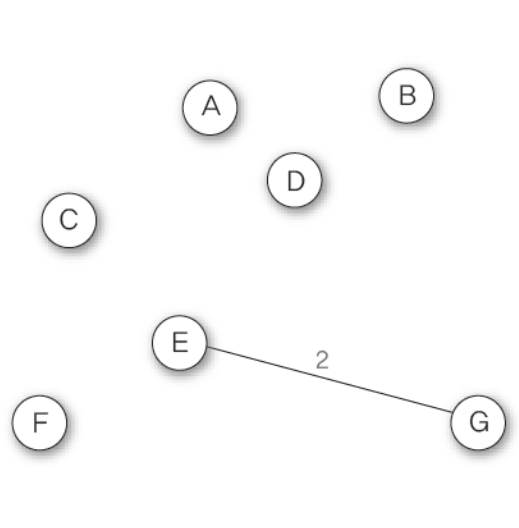
간선 수 : 11개

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 크루스칼 알고리즘 II

▶ 크루스칼 알고리즘 II 를 이용하여 G10의 최소 비용 신장 트리 만들기

① 가중치가 가장 낮은
간선 (E, G)를 삽입
→ 삽입한 간선 수는 한 개



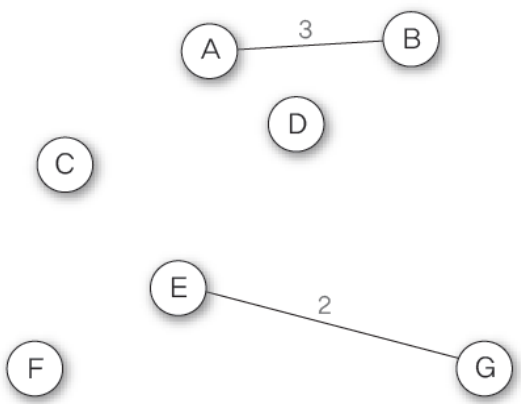
가중치	간선
2	(E, G)
3	(A, B)
4	(E, F)
5	(B, D)
6	(A, D)
8	(C, F)
9	(D, E)
10	(C, E)
12	(B, G)
14	(F, G)
17	(A, C)

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 크루스칼 알고리즘 II

▶ 크루스칼 알고리즘 II 를 이용하여 G10의 최소 비용 신장 트리 만들기

② 나머지 간선 중에서
가중치가 가장 낮은
간선 (A, B)를 삽입
→ 삽입한 간선 수는 두 개



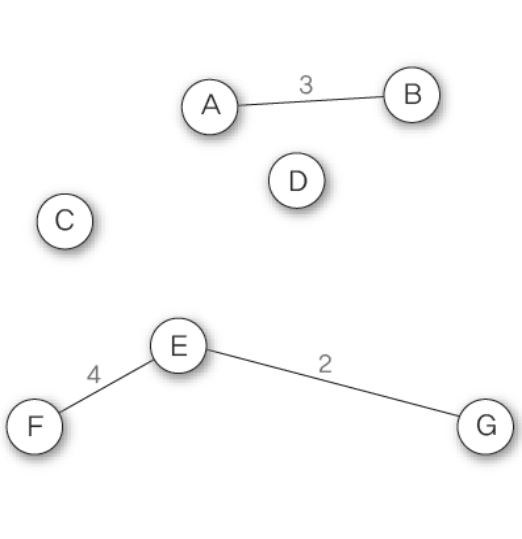
가중치	간선
2	(E, G)
3	(A, B)
4	(E, F)
5	(B, D)
6	(A, D)
8	(C, F)
9	(D, E)
10	(C, E)
12	(B, G)
14	(F, G)
17	(A, C)

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 크루스칼 알고리즘 II

▶ 크루스칼 알고리즘 II 를 이용하여 G10의 최소 비용 신장 트리 만들기

③ 나머지 간선 중에서
가중치가 가장 낮은
간선 (E, F)를 삽입
→ 삽입한 간선 수는 세 개



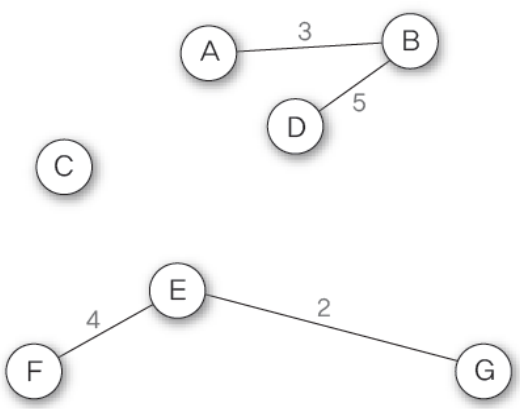
가중치	간선
2	(E, G)
3	(A, B)
4	(E, F)
5	(B, D)
6	(A, D)
8	(C, F)
9	(D, E)
10	(C, E)
12	(B, G)
14	(F, G)
17	(A, C)

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 크루스칼 알고리즘 II

▶ 크루스칼 알고리즘 II 를 이용하여 G10의 최소 비용 신장 트리 만들기

④ 나머지 간선 중에서
가중치가 가장 낮은
간선 (B, D)를 삽입
→ 삽입한 간선 수는 네 개



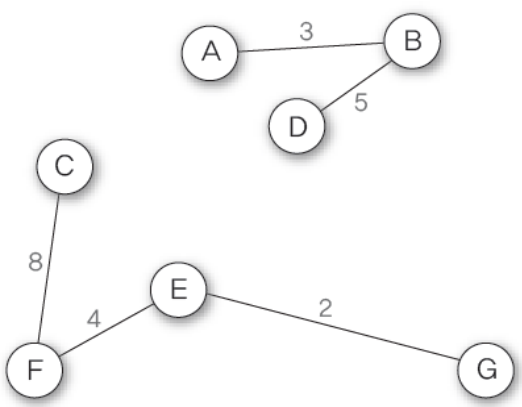
가중치	간선
2	(E, G)
3	(A, B)
4	(E, F)
5	(B, D)
6	(A, D)
8	(C, F)
9	(D, E)
10	(C, E)
12	(B, G)
14	(F, G)
17	(A, C)

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 크루스칼 알고리즘 II

▶ 크루스칼 알고리즘 II 를 이용하여 G10의 최소 비용 신장 트리 만들기

⑤ 나머지 간선 중에서 가중치가 가장 낮은 간선 (A, D)를 삽입하면 A-B-D-A의 사이클이 생성되므로 삽입할 수 없음, 이런 경우에는 그 다음으로 가중치가 낮은 간선 (C, F)를 삽입
→ 삽입한 간선 수는 다섯 개



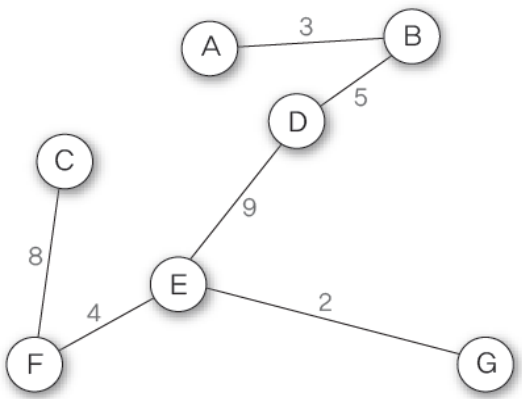
가중치	간선
2	(E, G)
3	(A, B)
4	(E, F)
5	(B, D)
6	(A, D)
8	(C, F)
9	(D, E)
10	(C, E)
12	(B, G)
14	(F, G)
17	(A, C)

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 크루스칼 알고리즘 II

▶ 크루스칼 알고리즘 II 를 이용하여 G10의 최소 비용 신장 트리 만들기

⑥ 나머지 간선 중에서
가중치가 가장 낮은
간선 (D, E)를 삽입
→ 삽입한 간선 수는 여섯 개



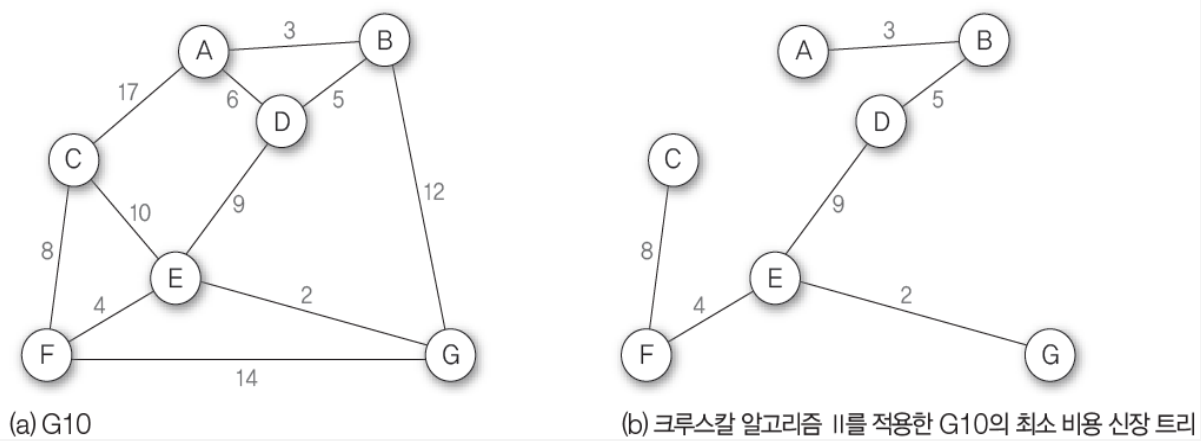
가중치	간선
2	(E, G)
3	(A, B)
4	(E, F)
5	(B, D)
6	(A, D)
8	(C, F)
9	(D, E)
10	(C, E)
12	(B, G)
14	(F, G)
17	(A, C)

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 크루스칼 알고리즘 II

- 크루스칼 알고리즘 II 를 이용하여 G10의 최소 비용 신장 트리 만들기

⑦ 현재 삽입한 간선이 여섯 개이므로 알고리즘 수행을 종료하면 신장 트리 완성



[그래프 G10을 크루스칼 알고리즘 II 으로 완성한 최소 비용 신장 트리]

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 프림 알고리즘

- 간선을 정렬하지 않고 하나의 정점에서 시작하여 트리를 확장해 나가는 방법
- 프림 알고리즘의 순서

❶ 그래프 G에서 시작 정점을 선택한다.

❷ 선택한 정점에 부속된 모든 간선 중에서 가중치가 가장 낮은 간선을 연결하여 트리를 확장한다.

❸ 이전에 선택한 정점과 새로 확장된 정점에 부속된 모든 간선 중에서 가중치가 가장 낮은 간선을 삽입한다.
단, 사이클을 형성하는 간선은 삽입할 수 없으므로 이런 경우에는 그 다음으로 가중치가 낮은 간선을 선택한다.

❹ 그래프 G에 간선이 n–1개 삽입될 때까지 ❸을 반복한다.

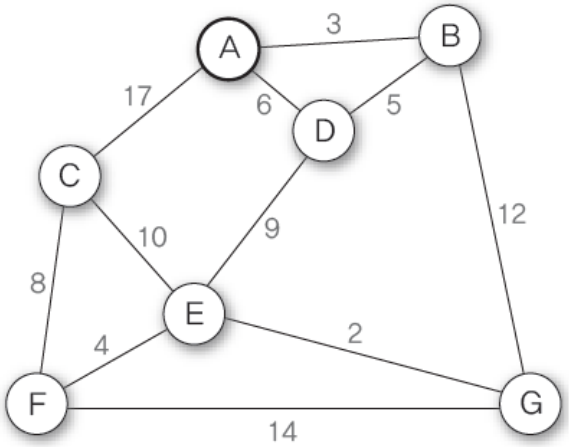
❺ 그래프 G의 간선이 n–1개가 되면 최소 비용 신장 트리가 완성된다.

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 프림 알고리즘

▶ 프림 알고리즘을 이용하여 그래프 G10의 최소 비용 신장 트리 만들기

⓪ 초기 상태
: 그래프 G10의 정점 중에서 정점 A를 시작 정점으로 선택



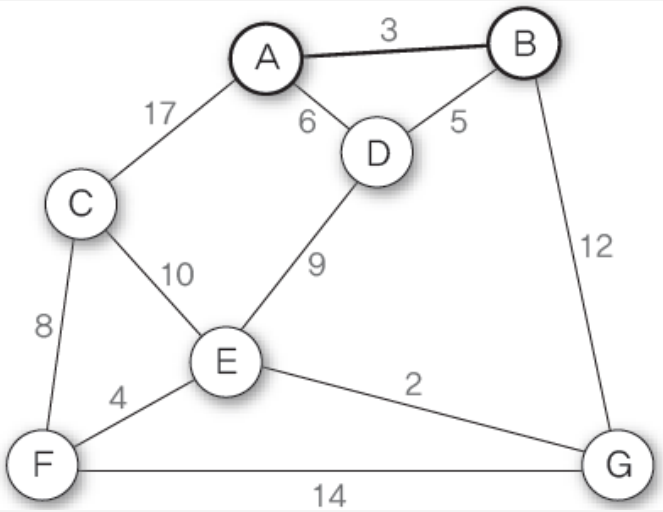
G10

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 프림 알고리즘

▶ 프림 알고리즘을 이용하여 그래프 G10의 최소 비용 신장 트리 만들기

① 정점 A에 부속된 간선 중에서 가중치가 가장 낮은 간선 (A, B)를 삽입하여 리를 확장 → 삽입한 간선 수는 한 개

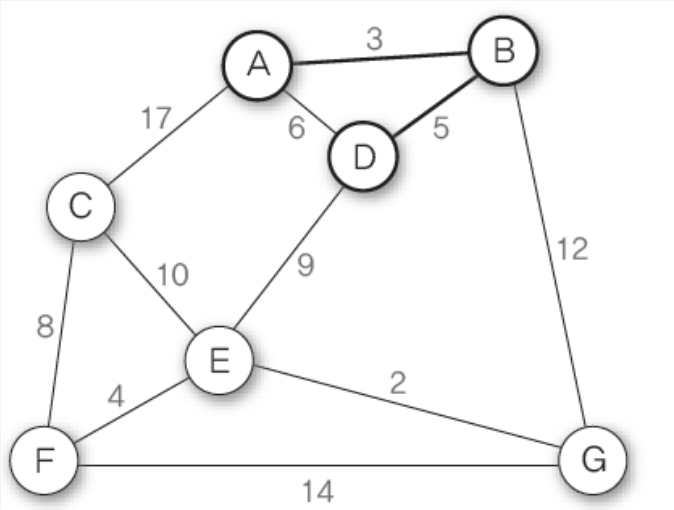


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 프림 알고리즘

▶ 프림 알고리즘을 이용하여 그래프 G10의 최소 비용 신장 트리 만들기

② 현재 확장된 트리의 정점 A, B에 부속된 간선 중에서 가중치가 가장 낮은 간선 (B, D)를 삽입하여 트리를 확장 → 삽입한 간선 수는 두 개

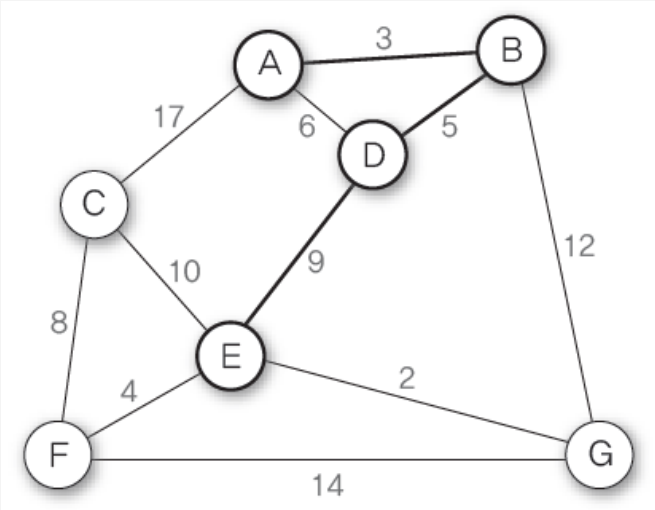


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 프림 알고리즘

▶ 프림 알고리즘을 이용하여 그래프 G10의 최소 비용 신장 트리 만들기

③ 현재 확장된 트리의 정점 A, B, D에 부속된 간선 중에서 가중치가 가장 낮은 간선인 (A, D)를 삽입하면 A-B-D-A의 사이클이 생성되므로 삽입할 수 없음, 따라서 그 다음으로 가중치가 낮은 간선 (D,E)를 삽입
→ 삽입한 간선 수는 세 개,
삽입 불가능한 간선은 (A, D)

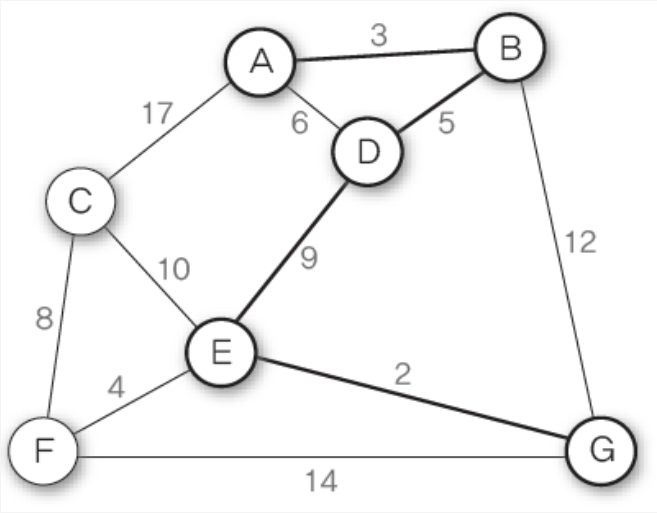


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 프림 알고리즘

▶ 프림 알고리즘을 이용하여 그래프 G10의 최소 비용 신장 트리 만들기

④ 현재 확장된 트리의 정점
A, B, D, E에 부속된 간선 중에서
가중치가 가장 낮은 간선 (E, G)를
삽입하여 트리를 확장
→ 삽입한 간선 수는 네 개,
삽입 불가능한 간선은 (A, D)

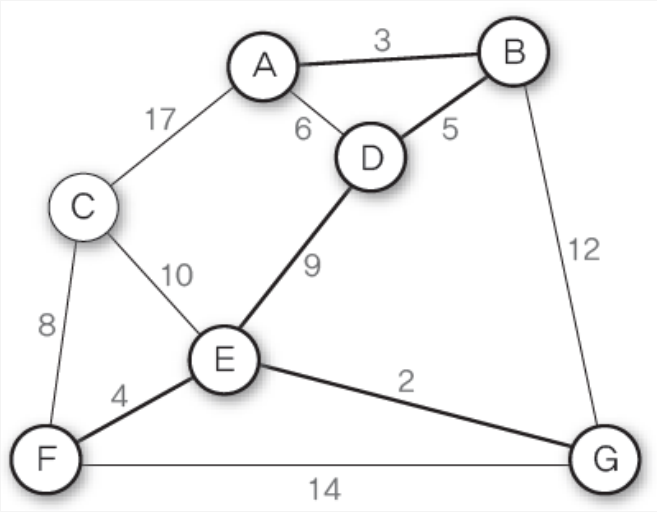


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 프림 알고리즘

▶ 프림 알고리즘을 이용하여 그래프 G10의 최소 비용 신장 트리 만들기

⑤ 현재 확장된 트리의 정점
A, B, D, E, G에 부속된 간선 중에서
가중치가 가장 낮은 간선 (E, F)를
삽입하여 트리를 확장
→ 삽입한 간선 수는 다섯 개,
삽입 불가능한 간선은 (A, D)

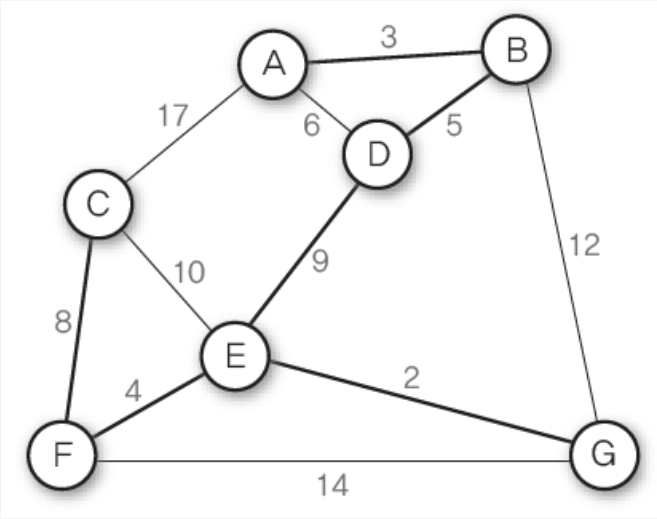


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 프림 알고리즘

▶ 프림 알고리즘을 이용하여 그래프 G10의 최소 비용 신장 트리 만들기

⑥ 현재 확장된 트리의 정점 A, B, D, E, F, G에 부속된 간선 중에서 가중치가 가장 낮은 간선 (C, F)를 삽입하여 트리를 확장
→ 삽입한 간선 수는 여섯 개, 삽입 불가능한 간선은 (A, D)

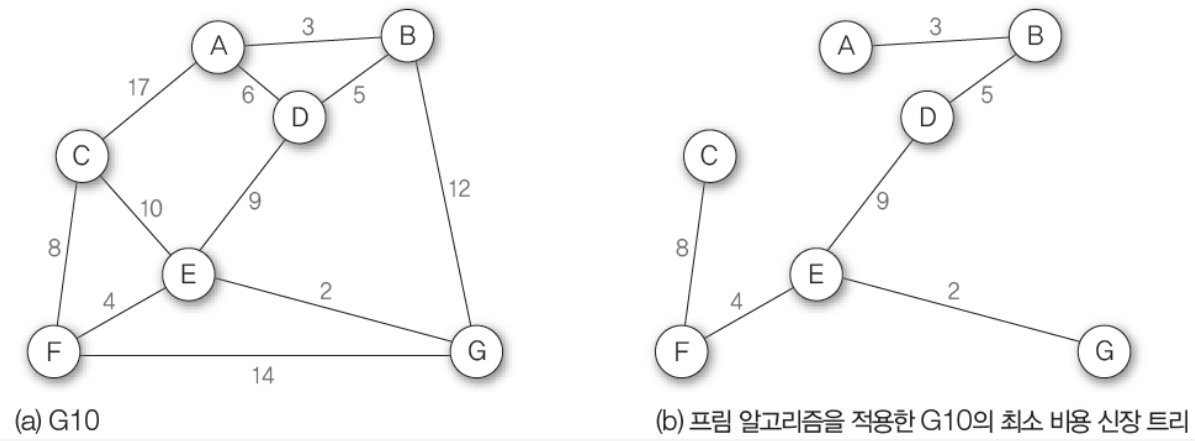


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 프림 알고리즘

- 프림 알고리즘을 이용하여 그래프 G10의 최소 비용 신장 트리 만들기

⑦ 현재 삽입한 간선 수가 여섯 개이므로 알고리즘 수행을 종료하면 신장 트리 완성



[그래프 G10을 프림 알고리즘으로 완성한 최소 비용 신장 트리]

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어