

This work is licensed under a Creative Commons “Attribution-NonCommercial-ShareAlike 3.0 Unported” license.

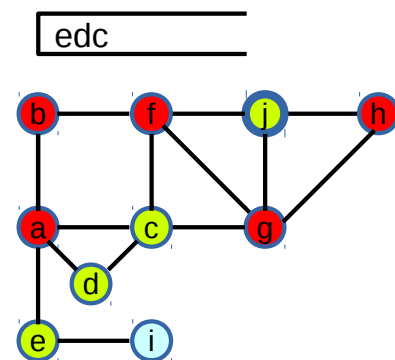
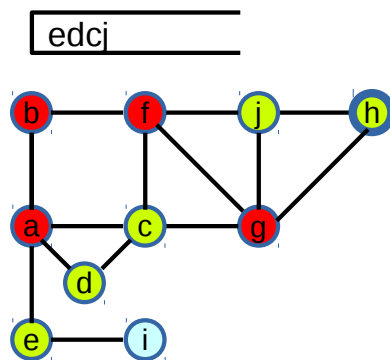
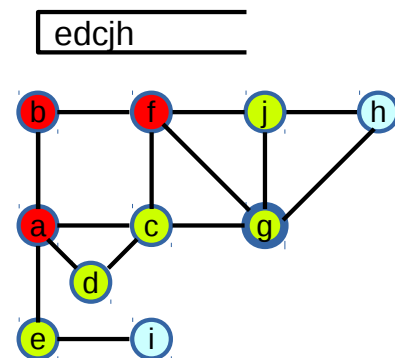
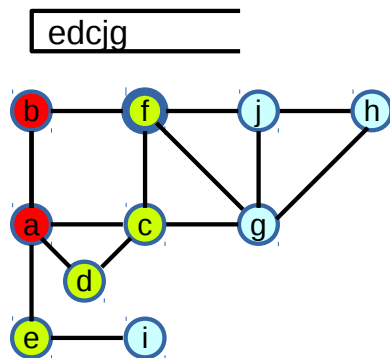
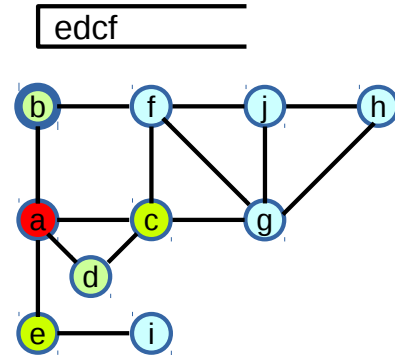
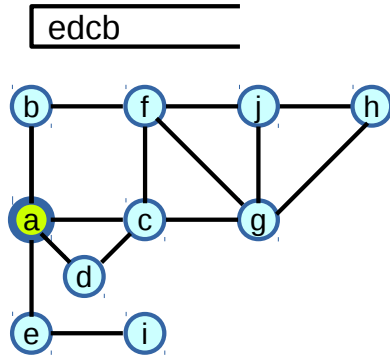


## 1 Graph Traversal

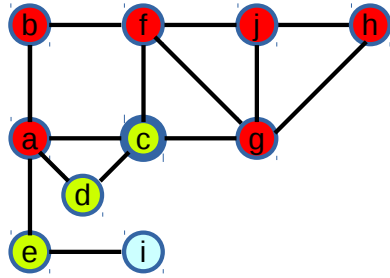
다음에 주어진 그래프에 대하여 DFS 알고리즘을 이용한 깊이 우선 순회와 BFS 알고리즘을 이용한 넓이 우선 순회를 다음과 같은 방법으로 구하시오.

### 1.1 Graph Traversal using the DFS Algorithm

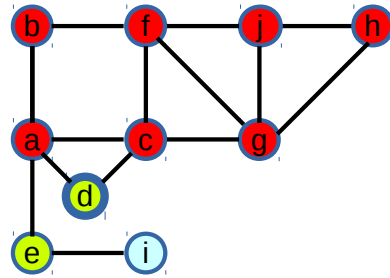
- 이전 단계의 그림에 있는 stack에서 한 개의 node를 pop한다.
- 이 node를 현 단계 그림의 current node로 표시한다.
- 이 node의 admissible arc에 연결된 node들을 현 단계의 그림에 있는 stack에 push한다.
- 현 단계의 current node와 push되는 node들은 다음 단계의 그림에서 각각 빨간색과 초록색으로 표시한다.
- stack에 push하는 순서는 pop을 연속적으로 수행 할 때 alphabet 순서로 pop이 되도록 역순으로 push한다고 가정한다. (예를 들어 e, a, c인 경우 push e, push c, push a이면 pop을 연속적으로 세 번 수행할 때 a, c, e의 순으로 나온다.)
- 처음에 stack은 node a가 push되어 있다고 가정한다.



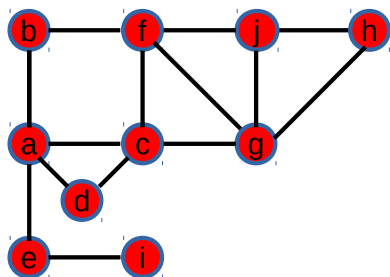
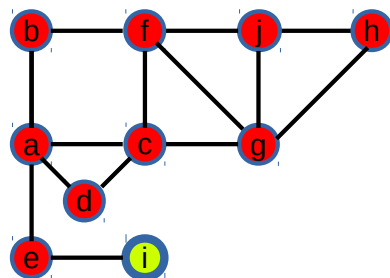
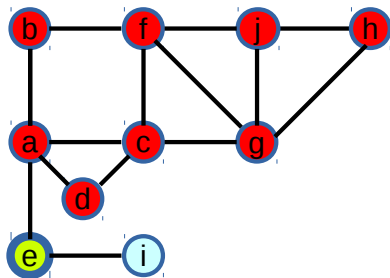
ed



e



i



## 1.2 Graph Traversal using the BFS Algorithm

- 이전 단계의 그림에 있는 queue에서 한 개의 node를 dequeue한다.
- 이 node를 현 단계 그림의 current node로 표시한다.
- 이 node의 admissible arc에 연결된 node들을 현 단계의 그림에 있는 queue에 enqueue한다.
- 현 단계의 current node와 push되는 node들은 다음 단계의 그림에서 각각 빨간색과 초록색으로 표시한다.
- queue에 enqueue하는 순서는 alphabet 순서로 한다고 가정한다. (예를 들어 e, a, c인 경우 enqueue a, enqueue c, enqueue e의 순으로 한다.)
- 처음에 queue에는 node a가 있다고 가정한다.

