

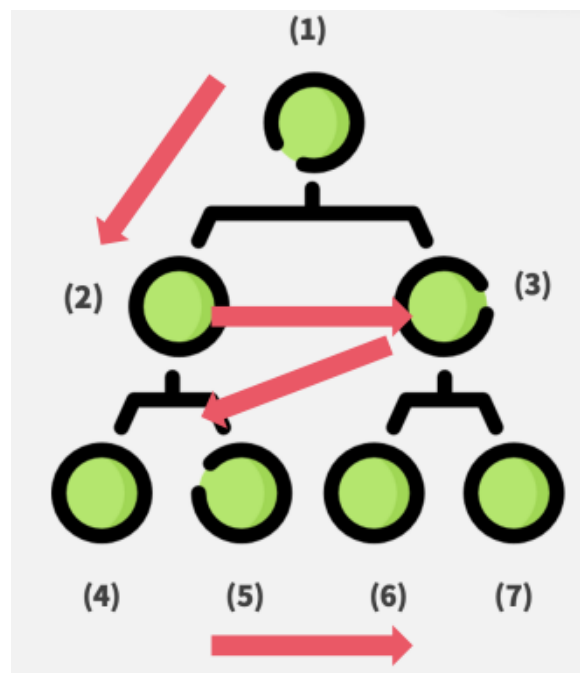
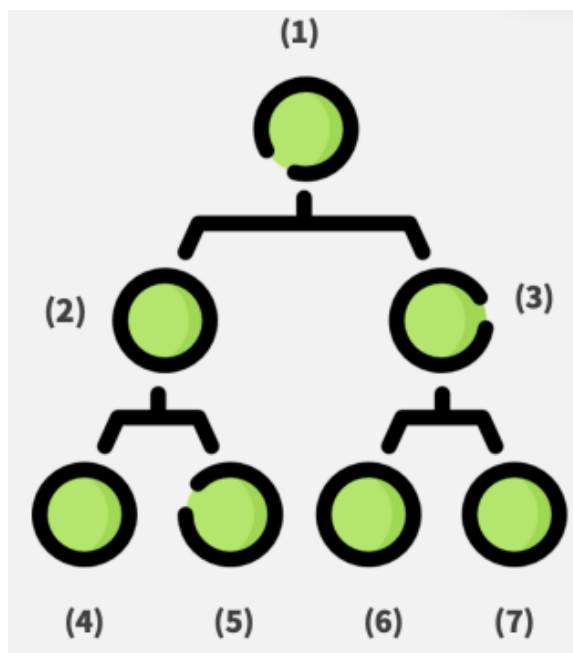


# 너비 우선 탐색 (BFS - Breadth First Search)

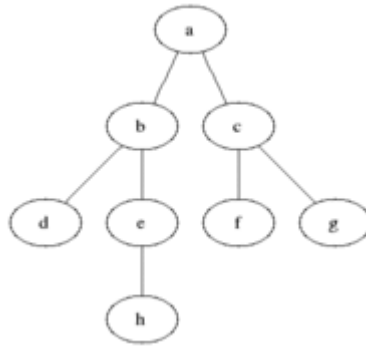
## BFS



너비 우선 탐색 : 루트, 임의의 노드에서 시작해 인접한 노드를 우선으로 탐색



너비 우선 탐색은 그래프, 트리에서 인접한 노드를 우선으로 탐색하는 방법입니다. 데이터를 트리 형태로 표현한다고 했을 때 트리의 **한 레벨을 다 탐색한 뒤에 다음 레벨을 탐색**한다는 뜻이죠.



## BFS 의사 코드

```

void BFS(){
    queue<int> q;
    q.push(시작 지점);
    visit[시작 지점] = true;    // 방문 체크를 할 bool 배열

    while(!q.empty()) {
        int cur = q.front();    // 이번에 방문한 노드
        q.pop();

        for(auto next : edge[cur]){
            if(visit[next]) continue;

            q.push(next);
            visit[next] = true;
        }
    }
}

```

큐를 사용하게 되면 자연스럽게 넓이 우선 탐색을 할 수 있게 됩니다.

## BFS의 특징

### 시작 노드에서 빠른 경로를 찾는게 중점

트리에서 간선의 가중치가 전부 같다고 할 때 **간선을 가장 적게 지나면서 원하는 노드에 도착하는 경로**를 찾는 데 유용하게 사용할 수 있습니다. 만약 가중치가 다른 경우 살짝 변형된 **다익스트라 알고리즘**을 사용해야 합니다.

### 메모리 초과가 날 가능성이 있다

BFS에서 사용하게 될 Queue에 무작정 데이터를 넣게 되면 메모리 초과가 일어날 가능성이 높습니다! 때문에 **메모리 초과** 가 난다면 문제가 **BFS의 성격에 맞는 문제인지, 큐에 이미 방문한 노드가 들어가지는 않는지** 살펴봅시다.

## 문제들이 입력값이 많은 경우

C++에서 cin을 사용해도 대부분 무리 없이 돌아가게 되지만 더 빠른 시간을 보고 싶으면 `printf`, `scanf` 나 입출력 속도를 높여주는 코드를 작성해야 합니다.

## 추천 문제

### 1697번: 숨바꼭질

수빈이는 동생과 숨바꼭질을 하고 있다. 수빈이는 현재 점  $N(0 \leq N \leq 100,000)$ 에 있고, 동생은 점  $K(0 \leq K \leq 100,000)$ 에 있다. 수빈이는 걷거나 순간이동을 할 수 있다. 만약, 수빈이의 위치가  $X$ 일 때 걷는

<https://www.acmicpc.net/problem/1697>

BAEKJOON  
ONLINE JUDGE

### 12851번: 숨바꼭질 2

수빈이는 동생과 숨바꼭질을 하고 있다. 수빈이는 현재 점  $N(0 \leq N \leq 100,000)$ 에 있고, 동생은 점  $K(0 \leq K \leq 100,000)$ 에 있다. 수빈이는 걷거나 순간이동을 할 수 있다. 만약, 수빈이의 위치가  $X$ 일 때 걷는

<https://www.acmicpc.net/problem/12851>

BAEKJOON  
ONLINE JUDGE

## 10026 ⇒ 기존 BFS + 추가적인 아이디어

### 10026번: 적록색약

적록색약은 빨간색과 초록색의 차이를 거의 느끼지 못한다. 따라서, 적록색약인 사람이 보는 그림은 아닌 사람이 보는 그림과는 좀 다를 수 있다. 크기가  $N \times N$ 인 그리드의 각 칸에 R(빨강), G(초록), B(파랑) 중

<https://www.acmicpc.net/problem/10026>

BAEKJOON  
ONLINE JUDGE

## 참고자료

### 너비 우선 탐색 - 위키백과, 우리 모두의 백과사전

너비 우선 탐색(Breadth-first search, BFS)은 맹목적 탐색방법의 하나로 시작 정점을 방문한 후 시작 정점에 인접한 모든 정점들을 우선 방문하는 방법이다. 더 이상 방문하지 않은 정점이 없을 때까지 방문하

W [https://ko.wikipedia.org/wiki/%EB%84%88%EB%B9%84\\_%EC%9A%B0%EC%84%A0\\_%ED%83%90%EC%83%89](https://ko.wikipedia.org/wiki/%EB%84%88%EB%B9%84_%EC%9A%B0%EC%84%A0_%ED%83%90%EC%83%89)

