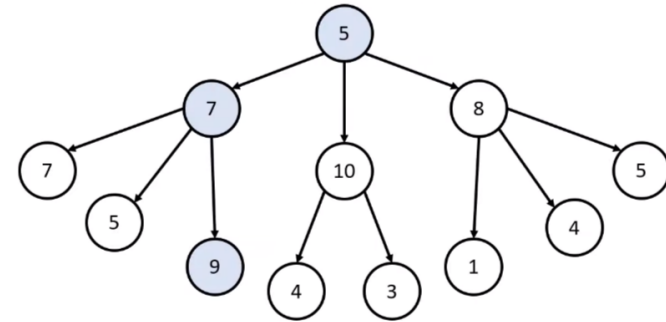


이러한 방법을 이용했을 때 문제에서 요구하는 최적의 해를 구할 수 있는지 검토하는 과정이 필요합니다.

## 그리디 알고리즘

- **[문제 상황]** 루트 노드부터 시작하여 거쳐 가는 노드 값의 합을 최대로 만들고 싶습니다.
  - Q. 최적의 해는 무엇인가요?



그리디 알고리즘

나동빈

위와 같이 트리가 구성되어 있을 때

우리는 이 그래프에서 어떤 식으로 이동하는게 최적의 해를 보장하는지 알 수 있죠.

5 -> 7 -> 9

이 순서대로 이동하면 노드 값의 합이 21로 가장 큰 경우가 됩니다.

## 그리디 알고리즘

나동빈

## 그리디 알고리즘

- 그리디 알고리즘(탐욕법)은 **현재 상황에서 지금 당장 좋은 것만 고르는 방법**을 의미합니다.
- 일반적인 그리디 알고리즘은 문제를 풀기 위한 최소한의 아이디어를 떠올릴 수 있는 능력을 요구합니다.
- 그리디 해법은 그 정당성 분석이 중요합니다.
  - 단순히 가장 좋아 보이는 것을 반복적으로 선택해도 최적의 해를 구할 수 있는지 검토합니다.

그리디 알고리즘

나동빈

나중에 배우게 될 크루스칼 알고리즘이나 다익스트라 알고리즘과 같이 잘 알려진 알고리즘을 제외하고

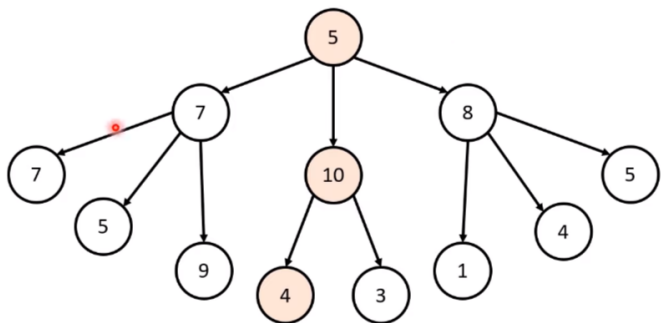
일반적으로 그리디 알고리즘일 출제되면 해당 문제를 풀기 위한 최소한의 아이디어를

적절히 떠올릴 수 있어야 문제가 풀리도록 출제되는 경우가 많습니다.

그리디 알고리즘은 탐욕적으로 현재 상황에서 지금 당장 좋은 것만 고르는 방법을 의미하는데

## 그리디 알고리즘

- **[문제 상황]** 루트 노드부터 시작하여 거쳐 가는 노드 값의 합을 최대로 만들고 싶습니다.
  - Q. 단순히 매 상황에서 가장 큰 값만 고른다면 어떻게 될까요?



그리디 알고리즘

나동빈

그렇다면 이렇게 그래프가 주어졌을 때

우리가 컴퓨터 프로그램을 이용해서 거쳐가는 노드 값의 합을 최대로 만들려고 할 때

이때 우리는 간단히 프로그램을 작성하는 방법으로

단순히 현재 위치에서 가장 큰 값만 선택하는 방법을 반복한다면 어떻게 될까요?

시작 노드인 root 노드 5 에서 출발한다고 했을 때 인접한 노드는 7, 10, 8 입니다.

이 경우 가장 큰 값을 가지는 10 으로 방문하게 됩니다.

이후 10 에서 이동할 수 있는 가장 큰 값인 4 로 이동하게 됩니다.

이 경우에는 5, 10, 4 를 더해서 총 19 만큼 합을 얻을 수 있으며 최적의 해인 21 보다는 낮은 값을 얻을 수 있겠죠.

**그리디 알고리즘은 이처럼 단순히 매 상황에서 가장 큰 값만 고르는 방식이라고 말할 수 있습니다.**

## 그리디 알고리즘

- 일반적인 상황에서 그리디 알고리즘은 최적의 해를 보장할 수 없을 때가 많습니다.
- 하지만 코딩 테스트에서의 대부분의 그리디 문제는 **탐욕법으로 얻은 해가 최적의 해가 되는 상황에서, 이를 추천할 수 있어야 풀리도록 출제됩니다.**

그리디 알고리즘

나동빈

그리디 알고리즘은 최적의 해를 보장할 수 없을 때가 많은데

실제로 여러분들이 개발할 때는 그리디 알고리즘을 써도 충분히 최적의 해에 가까운 값을 얻을 수 있거나

혹은 최적의 해를 얻을 수 있을 때 그리디 알고리즘을 사용하는 경우가 많은데

다만 코딩테스트에서는 일반적으로 어떠한 입력이 주어졌을 때

어떠한 출력이 나와야한다는 것을 출제자가 정해놓고 문제를 만드는 경우가 많기 때문에

만약 그리디 문제가 출제되면 탐욕법으로 얻은 해가 최적의 해가 되는 상황에서

단순히 그리디 알고리즘을 이용해도 이처럼 최적의 해를 얻을 수 있다는 것을 추천할 수 있어야

문제가 풀리도록 출제하는 경우가 많습니다.

탐욕법으로 분류되는 문제들은

탐욕법으로 얻은 해가 최적의 해가 되는 경우에 한해 문제를 출제하는 경우가 많습니다.