

---

# GREEDY ALGORITHM

---

Rolo - SkyJrCode

---

---

# 그리디 알고리즘의 정의 및 내용

---

- 미래를 고려하지 않고 현재 시점에 가장 좋은 선택을 하는 알고리즘
  - 관찰을 통해 탐색범위를 줄이는 알고리즘
  - 빠른 속도를 보장하지만, 최적해를 보장하지는 않는다.
-



---

# 이상적인 풀이 흐름

---

- 관찰을 통해 탐색범위를 줄이는 방법을 고안한다.
  - 탐색 범위를 줄여도 올바른 결과를 낸다는 사실을 수학적으로 증명한다.
  - 구현해서 문제를 통과한다.
-

---

# 관찰해야 할 사항

---

- 현재의 선택이 미래의 선택에 영향을 주지는 않는지?
  - 부분의 최적해가 모이면 전체의 최적해가 되는지?
  - 배수관계가 성립하는지?
    - 1월 9월 10원으로 18원 만들기.
-



---

# 동전 0 (백준 11047)

---

- N개의 동전들의 각각의 단위와 전체 총액 K가 주어졌을 때, K를 만들 수 있는 최소한의 동전 개수를 구하는 문제
  - 핵심 : 최소동전 개수
  - N개의 동전들 중에 가장 큰 단위의 금액을 먼저 선택
-



---

# 회의실 배정 (백준 1931)

---

- 하나의 회의실을 이용하고자 하는  $n$ 개의 사용요청을 입력받는다. 각 요청은 회의 시작 시간과 종료시간 정보를 제공하는데, 겹치지 않는 회의를 최대한 많이 진행한다면 몇 개까지 진행 할 수 있을 까?
  - 종료시간 기준으로 가능한 회의중에서 가장 먼저 끝나는 회의를 선택 한다.
-



---

# 그리디를 쓰는 이유?

---

- 다이나믹 프로그래밍은 모든 최적해를 검색하는 경향이 있다. 대신 속도가 너무 느려서 문제가 생기는데, 빠른 시일 내에 정답 비스무리한 것을 찾고 싶으면 그리디로 문제를 푸는 것이 낫다.
  - 최적해가 요청되지 않는 곳에 사용 할 수 있다.
  - 코딩 테스트에는 최적해가 보장되는 경우에만 그리디를 쓰면 된다.
  - 현실 세계에서는 완벽한 정답이 아닌 근사치 정도만 답이 나와도 만족스러운 경우에 성능을 개선하기 위해서 사용한다.
-

---

## ■ Q & A

---