

청년 AI 아카데미 28기 알고리즘 실습

그리디기법 & 동적계획법

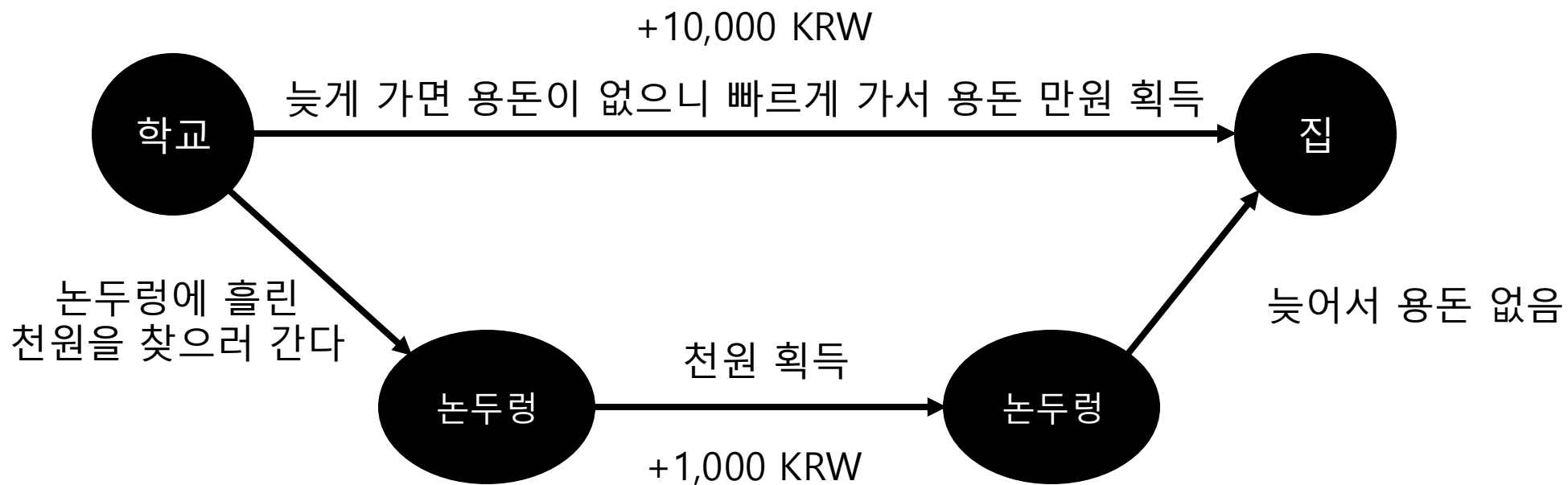
Today

- 그리디 기법
 - 세금 징수
- 다이나믹 프로그래밍
 - 피보나치 수열 2 (1차원 DP)
 - 세계암기대회 (2차원 DP)

Greedy Method?

Greedy의 기본: 현재 상황만 보고 가장 이득을 취할 수 있는 방향으로 움직인다!

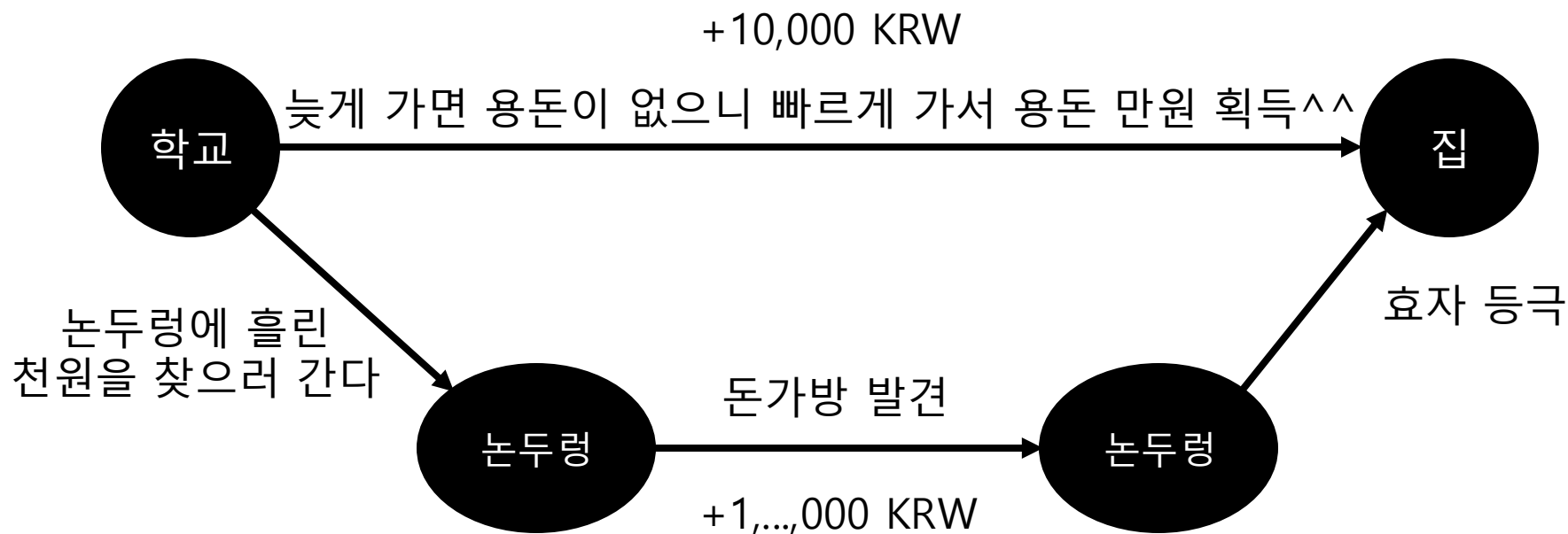
- 특정 상황에서 최적의 값을 찾는다.
- 알고리즘이 단순하고 빠르기 때문에 근사 알고리즘을 설계할 때 자주 채택한다.



Greedy Method?

Greedy의 기본: 현재 상황만 보고 가장 이득을 취할 수 있는 방향으로 움직인다!

- 특정 상황에서 최적의 값을 찾는다.
- 알고리즘이 단순하고 빠르기 때문에 근사 알고리즘을 설계할 때 자주 채택한다.



01. 세금 징수

특정 금액이 주어졌을 때, 해당 금액을 만드는 동전(지폐)의 최소 개수를 구합니다.

동전 단위 : 50000, 10000, 5000, 1000, 500, 100

예시 : 74100



VS



× 741

01. 세금 징수

특정 금액이 주어졌을 때, 해당 금액을 만드는 동전(지폐)의 최소 개수를 구합니다.

Greedy : 작은 단위에서부터? **큰 단위에서부터?**

*Hint: 하나씩 빼기보다는 나누기와 나머지를 활용!

$$9 = 5 + 1 + 1 + 1 + 1$$

$$9 // 5 = 1$$

$$9 \% 5 = 4$$

01. 세금 징수

주의!!

그리디 기법이 최적값을
항상 도출하는지 생각해야한다.

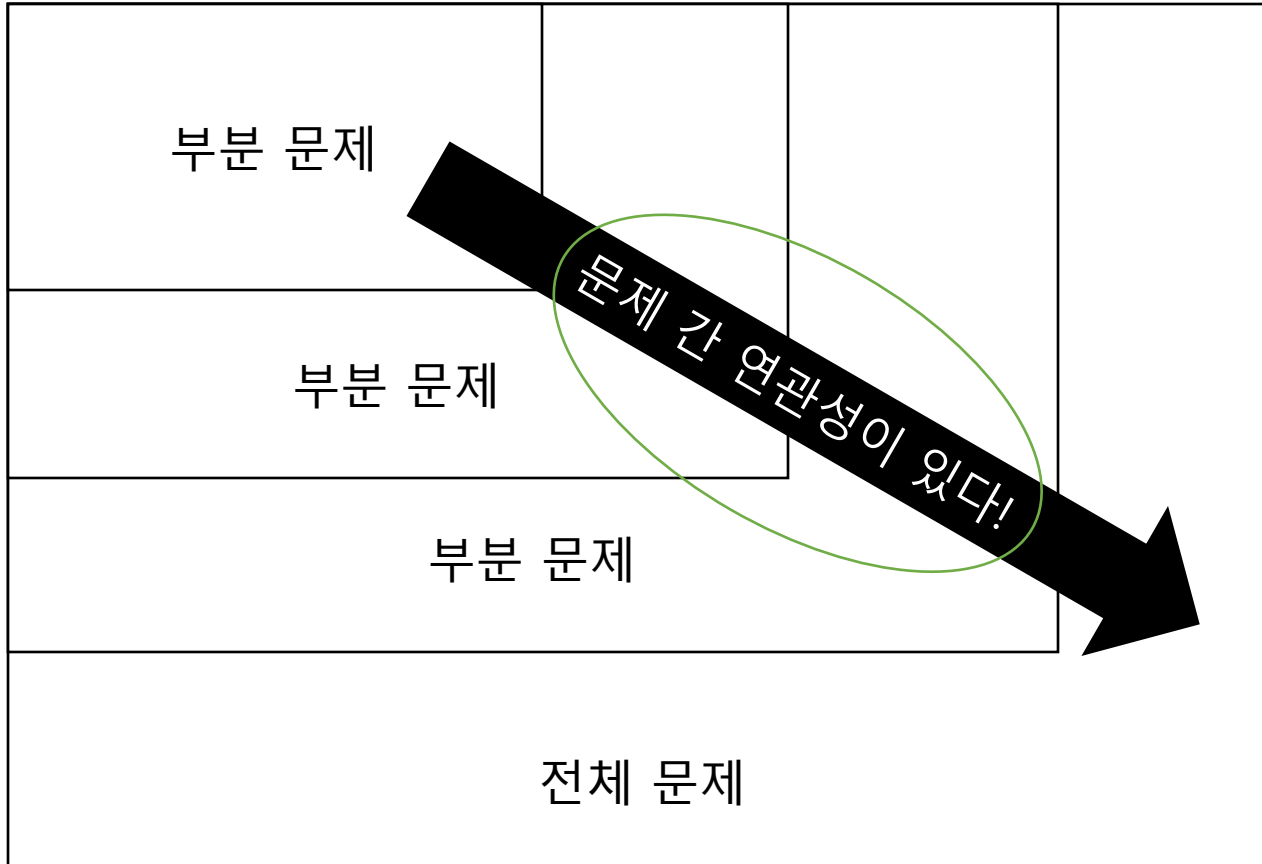
Ex1: 사용 가능한 동전 [100,500]

800원을 지불하는 방법 $\rightarrow 800 = 500 + 100 \times 3$ (4개)

Ex2: 사용 가능한 동전 [100,400,500]

800원을 지불하는 방법 $\rightarrow 800 = 400 \times 2$ (2개)

Dynamic Programming?



작은 부분 문제를 풀고, 그것들을
이용해 큰 전체 문제를 해결!

- Bottom Up 방식
- 분할한 문제 간의 **연관성 O**

c.f. Divide & Conquer

- Top Down 방식
- 분할한 문제 간의 **연관성 X**

Dynamic Programming?

1. 부분 문제를 명확하게 정의합니다.

- 원래 문제에서 보통 데이터의 크기만 줄이는 경우로 생각
- 예외적인 케이스도 존재 → 기존 부분 문제 + 제한 조건을 추가

2. 원래 답의 위치를 파악합니다.

- 부분 문제가 곧 답의 힌트!
- 부분 문제의 해답을 모아둘 배열 등을 생성 가능
- 부분 문제가 틀림을 알아내는 데에도 주요한 포인트

사람이 해결해야 함!



3. 재귀 식(점화 식)을 부분 문제의 정의와 수학적 논리에 따라 잘 세웁니다.

- 부분 문제를 해결하기 위해선, 더 작은 부분 문제들의 답을 이용
- Backward Analysis: 이 문제의 답이 어디서 올 수 있었는가를 분석
- 식에 나오게 되는 변수들을 통해 반복 문을 어떻게 해야 할지 분석
- 식이 제대로 세워지지 않는다면, 1번으로 돌아감

4. 기저 조건(초기값)을 세웁니다.

- 문제 조건으로 주어진 초기 값
- 3번 식에서는 값을 구할 수 없는 예외적인 경우

02. 피보나치 수열 2

n번째 피보나치 수를 계산합니다.

Dynamic Programming의 핵심은 대부분 펜과 노트에서 완성됩니다.

$$\text{FIB}[1] = 1$$

$$\text{FIB}[2] = 1$$

$$\text{FIB}[3] = 2$$

....

$$\text{FIB}[n] = \text{FIB}[n-1] + \text{FIB}[n-2]$$

02. 피보나치 수열 2

1. 부분 문제를 명확하게 정의
 - $FIB[i]$: i 번째 피보나치 수
2. 원래 답의 위치
 - $FIB[n]$
3. 재귀 식(점화 식)
 - $FIB[i] = FIB[i-1] + FIB[i-2]$
4. 기저 조건(초기값)
 - $i=1$ 일 때, $FIB[i] = 1$
 - $i=2$ 일 때, $FIB[i] = 1$

03. 세계 암기대회

태훈이가 오른쪽 아래로 가면서 읽을 수 있는 가장 적은 점수를 계산해주세요!

1	0	1	0	1
0	1	1	1	0
0	1	0	1	1

03. 세계 암기대회

각 칸으로 가면서 읽을 수 있는 최저 점수를 저장한다면?

1	0	1	0	1
0	1	1	1	0
0	1	0	1	1



1	1	2	2	3
1	2	2	?	?
1	2	?	?	??

03. 세계 암기대회

$mp(i, j)$: (i,j)까지 가면서 잃는 최소 점수
 $p(i, j)$: (i,j)에 있는 점수

$$mp(i, j) = \begin{cases} p(i, j) & \text{if } i = 0, j = 0 \\ mp(i, j - 1) + p(i, j) & \text{elif } i = 0, j > 0 \\ mp(i - 1, j) + p(i, j) & \text{elif } i > 0, j = 0 \\ \min(mp(i - 1, j), mp(i, j - 1), mp(i - 1, j - 1)) + p(i, j) & \text{elif } i > 0, j > 0 \end{cases}$$