

# 알고리즘 - 그리디

# Greedy

## 탐욕 알고리즘

### 탐욕 알고리즘 (Greedy)

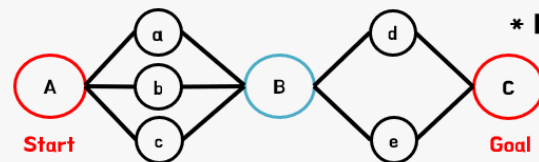
- 탐욕 알고리즘은 최적해를 구하는 데 활용
- 매 순간 최적의 선택으로 **최종적인 해가 되도록 하는 알고리즘**
- 최적 부분 구조(optimal substructure)
  - 문제의 최적해가 부분 문제에서도 최적해가 된다.
- 탐욕적 선택 속성 (greedy choice property)
  - **앞의 선택이 이후의 선택에 영향을 주지 않는다.**
- 그리디 알고리즘은 특성상 실행 속도가 매우 빠르다.
- 문제 출제 빈도는 낮지만, 그리디한 속성을 찾는 것이 어렵기 때문에 난이도는 있는 편

### 최적부분구조(Optimal Substructure)

전체 문제의 해가 부분문제로 구할 수 있음  
=> ex.  $F(A,C) = F(A,B) + F(B,C)$

### 탐욕적 선택 속성(Greedy Choice Property)

지역적 최적 선택이 전체 문제의 해에 포함  
=> ex.  $F(A,B) = \max(F(A,a), F(A,b), F(A,c))$



\*  $F(A, C)$  : A와 C의 최단경로

## 예제 : 거스름돈 문제

- 당신은 음식점의 계산을 도와주는 점원이다. 카운터에는 거스름돈으로 사용할 500원, 100원, 50원, 10원짜리 동전이 무한히 존재한다. 손님에게 거슬러 줘야 할 돈이 N원일 때 거슬러 줘야할 동전의 최소 개수를 구하라. (단, N은 항상 10의 배수)

- 예시 입출력

입력  
1260

출력  
6

- 전체문제를 부분문제의 식으로 표현해볼 것

## 예제 : 거스름돈 문제

- 당신은 음식점의 계산을 도와주는 점원이다. 카운터에는 거스름돈으로 사용할 500원, 100원, 50원, 10원짜리 동전이 무한히 존재한다. 손님에게 거슬러 줘야 할 돈이 N원일 때 거슬러 줘야할 동전의 최소 개수를 구하라. (단, N은 항상 10의 배수)

- 예시 입출력

입력  
1260

출력  
6

- 전체문제를 부분문제의 식으로 표현해볼 것

우선, 문제에서 제시한 내용을 확인한다.

우리는 500, 100, 50, 10원짜리 동전을 손님에게 거슬러 줘야하고, 동전 개수를 최소로 해야 한다.



## 예제 : 거스름돈 문제

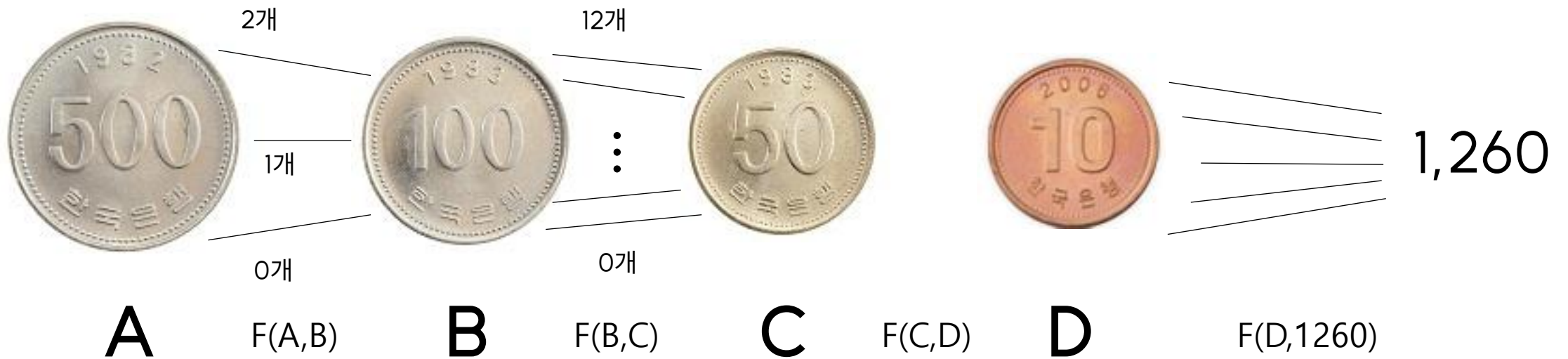
- 당신은 음식점의 계산을 도와주는 점원이다. 카운터에는 거스름돈으로 사용할 500원, 100원, 50원, 10원짜리 동전이 무한히 존재한다. 손님에게 거슬러 줘야 할 돈이 N원일 때 거슬러 줘야할 동전의 최소 개수를 구하라. (단, N은 항상 10의 배수)

문제를 풀기 위한 부분문제의 식은 다음과 같이 표현할 수 있다.

F를 각각의 동전을 선택한 개수라고 한다면,

$$F(A, 1260) = \min ( 500 * F(A, B) + 100 * F(B, C) + 50 * F(C, D) + 10 * F(D, 1260) )$$

와 같이 표현할 수 있다.



## 예제 : 거스름돈 문제

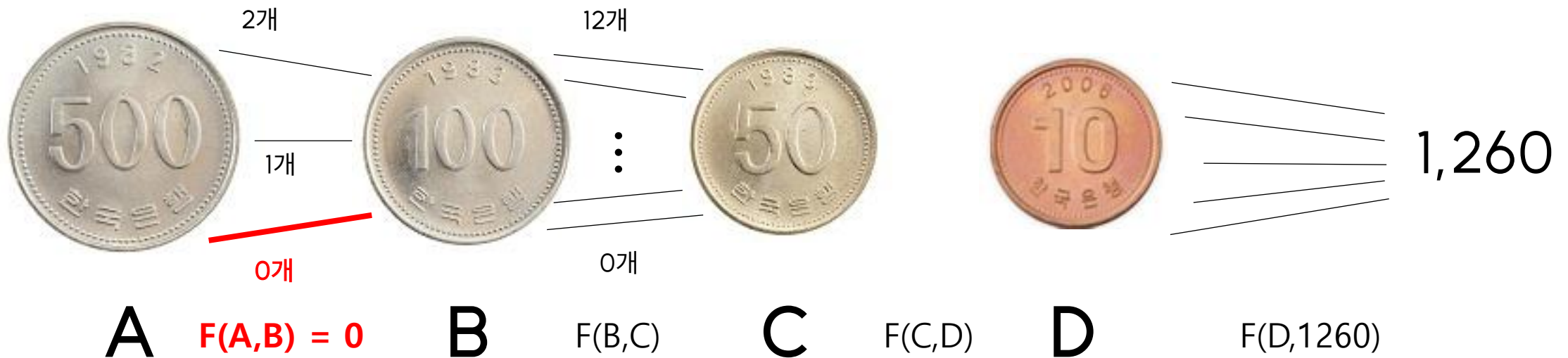
- 당신은 음식점의 계산을 도와주는 점원이다. 카운터에는 거스름돈으로 사용할 500원, 100원, 50원, 10원짜리 동전이 무한히 존재한다. 손님에게 거슬러 줘야 할 돈이 N원일 때 거슬러 줘야할 동전의 최소 개수를 구하라. (단, N은 항상 10의 배수)

문제를 풀기 위한 부분문제의 식은 다음과 같이 표현할 수 있다.

F를 각각의 동전을 선택한 개수라고 한다면,

$$F(A, 1260) = \min ( 500 * F(A, B) + 100 * F(B, C) + 50 * F(C, D) + 10 * F(D, 1260) )$$

500원짜리 동전을 0개 선택하는 경우를 생각해보자,



## 예제 : 거스름돈 문제

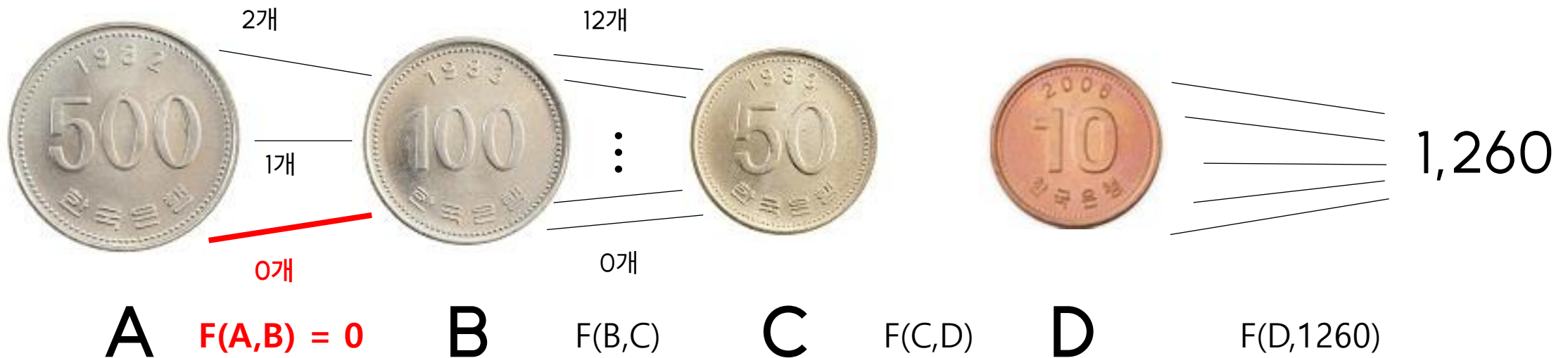
- 당신은 음식점의 계산을 도와주는 점원이다. 카운터에는 거스름돈으로 사용할 500원, 100원, 50원, 10원짜리 동전이 무한히 존재한다. 손님에게 거슬러 줘야 할 돈이 N원일 때 거슬러 줘야할 동전의 최소 개수를 구하라. (단, N은 항상 10의 배수)

문제를 풀기 위한 부분문제의 식은 다음과 같이 표현할 수 있다.

F를 각각의 동전을 선택한 개수라고 한다면,

$$F(A, 1260) = \min ( 500 * F(A, B) + 100 * F(B, C) + 50 * F(C, D) + 10 * F(D, 1260) )$$

당연하게도, 사람은 직관적으로 100원짜리 동전의 개수나, 50원짜리, 10원짜리 동전을 많이 써야 거슬러 줄 수 있다는 것을 알 수 있다.



## 예제 : 거스름돈 문제

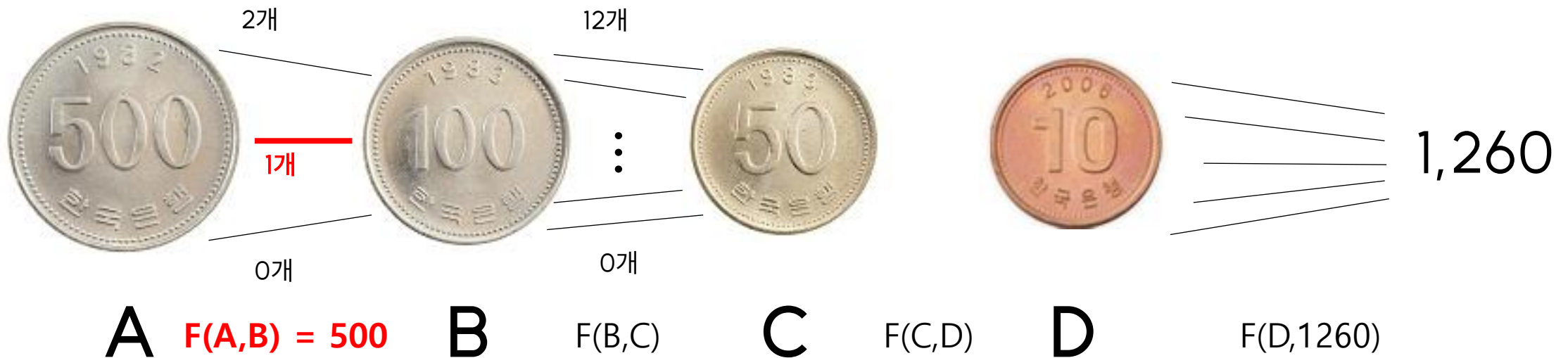
- 당신은 음식점의 계산을 도와주는 점원이다. 카운터에는 거스름돈으로 사용할 500원, 100원, 50원, 10원짜리 동전이 무한히 존재한다. 손님에게 거슬러 줘야 할 돈이 N원일 때 거슬러 줘야할 동전의 최소 개수를 구하라. (단, N은 항상 10의 배수)

문제를 풀기 위한 부분문제의 식은 다음과 같이 표현할 수 있다.

F를 각각의 동전을 선택한 개수라고 한다면,

$$F(A, 1260) = \min ( 500 * F(A, B) + 100 * F(B, C) + 50 * F(C, D) + 10 * F(D, 1260) )$$

그럼, 이번엔 500원짜리 동전을 1개 선택하는 경우를 생각해보자,





## 예제 : 거스름돈 문제

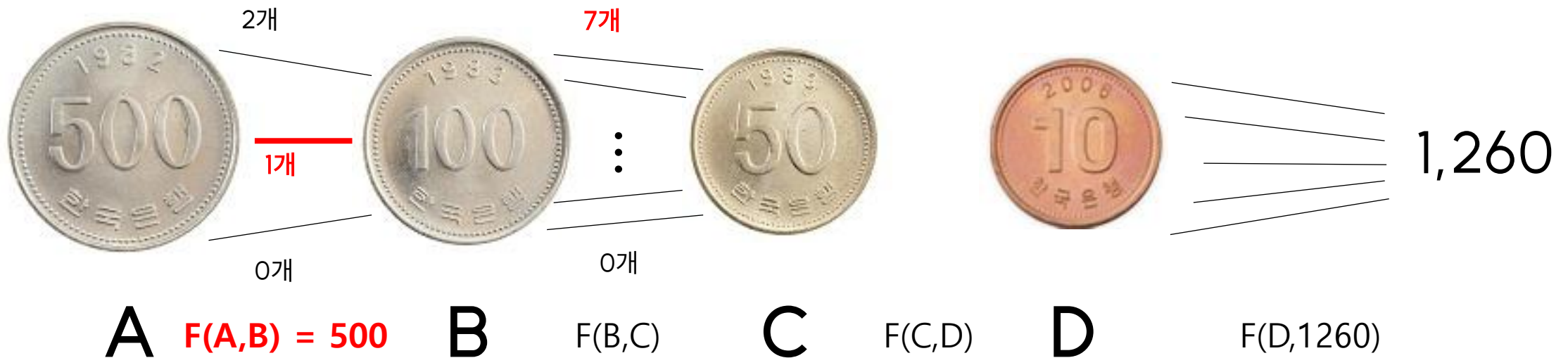
- 당신은 음식점의 계산을 도와주는 점원이다. 카운터에는 거스름돈으로 사용할 500원, 100원, 50원, 10원짜리 동전이 무한히 존재한다. 손님에게 거슬러 줘야 할 돈이 N원일 때 거슬러 줘야할 동전의 최소 개수를 구하라. (단, N은 항상 10의 배수)

문제를 풀기 위한 부분문제의 식은 다음과 같이 표현할 수 있다.

F를 각각의 동전을 선택한 개수라고 한다면,

$$F(A, 1260) = \min ( 500 * F(A, B) + 100 * F(B, C) + 50 * F(C, D) + 10 * F(D, 1260) )$$

500원짜리 동전을 하나 씬으로 인해서, 100원짜리 동전을 선택할 수 있는 최대 개수가 7개로 줄어든 것을 알 수 있다. 100원짜리부터 0개를 선택한다고 해도, 사용해야 할 10원짜리 동전 개수는 많이 줄어들게 된다. 126개 -> 76개 + 1



## 예제 : 거스름돈 문제

- 당신은 음식점의 계산을 도와주는 점원이다. 카운터에는 거스름돈으로 사용할 500원, 100원, 50원, 10원짜리 동전이 무한히 존재한다. 손님에게 거슬러 줘야 할 돈이 N원일 때 거슬러 줘야할 동전의 최소 개수를 구하라. (단, N은 항상 10의 배수)

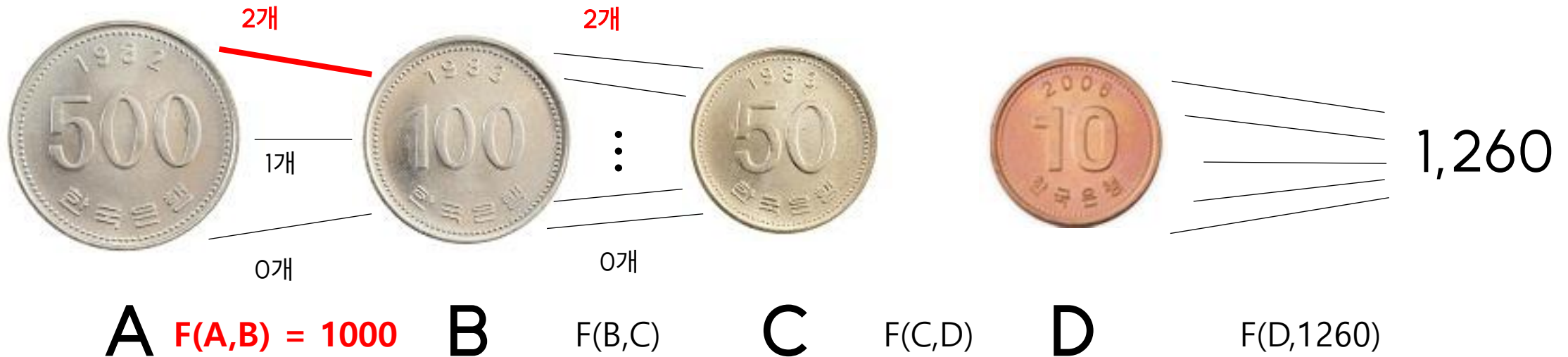
문제를 풀기 위한 부분문제의 식은 다음과 같이 표현할 수 있다.

F를 각각의 동전을 선택한 개수라고 한다면,

$$F(A, 1260) = \min ( 500 * F(A, B) + 100 * F(B, C) + 50 * F(C, D) + 10 * F(D, 1260) )$$

이렇게 해서 하나하나씩 확인해 볼 수도 있지만, 자연스럽게 우리는 단위가 큰 동전의 개수를 많이 가져가면 가져갈수록, 단위가 작은 동전의 개수 사용량이 줄어든다는 것을 확인할 수 있다.

단위가 큰 동전을 가장 많이 사용한다는 탐욕적 선택 속성이 동전의 개수를 최소화하는 전체 문제의 부분적인 해가 되는 상황이라고 할 수 있다.



## 예제 : 거스름돈 문제

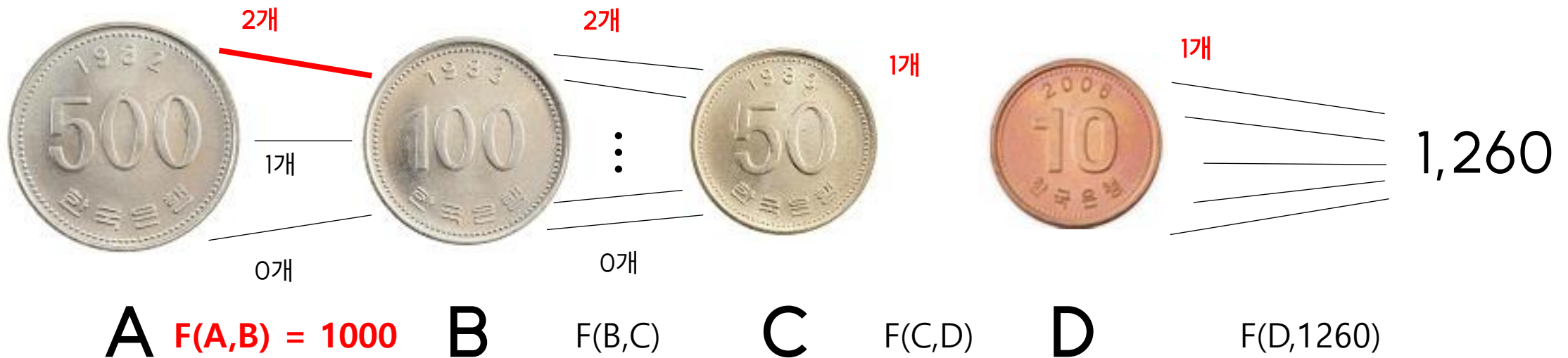
- 당신은 음식점의 계산을 도와주는 점원이다. 카운터에는 거스름돈으로 사용할 500원, 100원, 50원, 10원짜리 동전이 무한히 존재한다. 손님에게 거슬러 줘야 할 돈이 N원일 때 거슬러 줘야할 동전의 최소 개수를 구하라. (단, N은 항상 10의 배수)

문제를 풀기 위한 부분문제의 식은 다음과 같이 표현할 수 있다.

F를 각각의 동전을 선택한 개수라고 한다면,

$$F(A, 1260) = \min ( 500 * F(A, B) + 100 * F(B, C) + 50 * F(C, D) + 10 * F(D, 1260) )$$

따라서, 이렇게 큰 동전부터 하나씩 선택을 해 나가면, 최소 동전 개수는 2 + 2 + 1 + 1로, 6개가 된다.



## 예제 : 거스름돈 문제

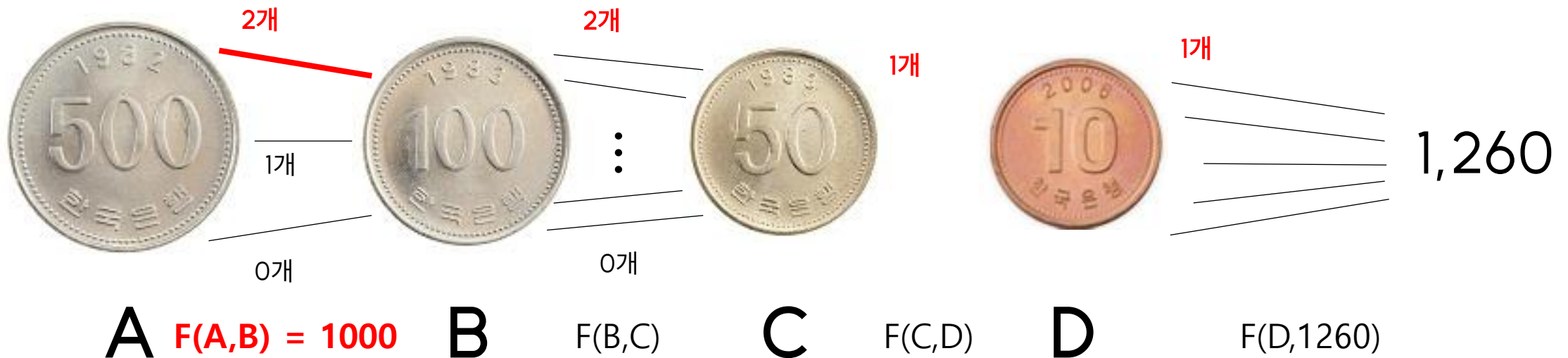
- 당신은 음식점의 계산을 도와주는 점원이다. 카운터에는 거스름돈으로 사용할 500원, 100원, 50원, 10원짜리 동전이 무한히 존재한다. 손님에게 거슬러 줘야 할 돈이 N원일 때 거슬러 줘야할 동전의 최소 개수를 구하라. (단, N은 항상 10의 배수)

문제를 풀기 위한 부분문제의 식은 다음과 같이 표현할 수 있다.

이를 코드로 한 번 구현해보자.

전체 로직을 말로 설명하면 다음과 같다.

- 1) 가장 큰 단위의 동전으로 거슬러 줄 수 있는 최대 개수를 구한다.
- 2) 그것을 정답에 더하고, 남은 금액을 그 다음 단위의 동전으로 1의 과정을 반복한다.
- 3) 그렇게 10원짜리 동전까지 진행한 뒤, answer를 반환한다.



**실전 : 체육복 (Lv.1)**

<https://school.programmers.co.kr/learn/courses/30/lessons/42862>

**실전 : 조이스틱 (Lv.2)**

<https://school.programmers.co.kr/learn/courses/30/lessons/42860>

**실전 : 큰 수 만들기 (Lv.2)**

<https://school.programmers.co.kr/learn/courses/30/lessons/42883>

**실전 : 구명보트 (Lv.2)**

<https://school.programmers.co.kr/learn/courses/30/lessons/42885>