



파이썬과 함께하는 이세계 코딩 생활

(한 주 쉬고) 8주차 : 그리디



그리디 (Greedy, 탐욕법, 욕심쟁이 문제)

greedy

미국·영국['gri:di]  영국식 

형용사

탐욕스러운, 욕심 많은

You **greedy** pig! You've already had two helpings! 

이 욕심 많은 돼지 같으니라고! 넌 벌써 두 번이나 덜어 먹었잖아!

[영어사전 결과 더보기](#)



그리디 문제의 특징

- 최적화 문제 (나올 수 있는 여러가지 해들 중에, 가장 좋은 해를 찾는다)
- 입력 데이터들의 관계를 고려하지 않고,
각 단계에서 최적의 데이터(최대값 또는 최소값)를 선택한다.
- 이때, 한 번 선택하면 다음 단계에서 그 선택을 취소할 수 없다.
- 따라서 그리디 알고리즘은 단순한 편이며, 최적의 데이터를 선택한다는 간단한 아이디어를 떠올릴 수 있는 능력을 요구한다. (암기 X)

그리디 알고리즘이 적용되기 위한 조건

1. Greedy choice property, 탐욕적 선택 조건

: 탐욕적 선택을 했을 때, 전체 문제의 최적해를 반드시 구할 수 있어야 한다.

이전의 선택이 이후의 선택에 영향을 주지 않아야 한다.

2. Optimal sub-structure, 최적 부분 구조 조건

: [전체 문제의 최적해 = 탐욕적 선택 + 하위 문제의 최적해]

주어진 전체 문제에 대한 최적해를 구하려고 할 때, 좀 더 작은 부분 문제들에 대한 최적해를 찾은 후, 그걸 결합한 게 전체 문제의 최적해가 되어야 한다.

이 조건들이 성립하지 않는 경우, 그리디 알고리즘으로 최적해를 구하지 못한다.



예시 (그리디 알고리즘을 모든 문제에 적용하면 안 된다 π)

문제 : 5월 22일에 제 1회 인터페이스 컨퍼런스, 인터퍼런스를 진행한다 🍷

신입부원인 김인페는 가장 오랜 시간 동안 컨퍼런스에 참여하고 싶다. 총 6개의 발표가 있고 몇 분 동안 진행되는지는 발표 시작 전까지는 비밀이라고 할 때, 김인페는 어떤 발표들을 선택하면 될까?

아무 조건이 없는 경우 → 그리디

10 40 30 20 50 100

발표를 딱 3개만 들을 수 있는 경우 → 그리디 X

10 40 30 20 50 100

발표를 연속으로는 못 듣는 경우 → 그리디 X

10 40 30 20 50 100

거스름돈

김인페는 잡화점의 알바생이다.

이곳은 카드를 사용하지 않아서, 손님이 물건을 구매하면 동전으로 거스름돈을 줘야한다.
손님에게 거슬러 줘야할 돈이 N 원일 때, **거스름돈으로 주는 동전의 최소 개수**를 구하라!

단, N 원은 항상 10의 배수이고, 동전은 500원, 100원, 50원, 10원이 있다.



거스름돈

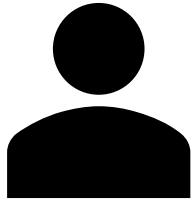
김인페는 잡화점의 알바생이다.

이곳은 카드를 사용하지 않아서, 손님이 물건을 구매하면 동전으로 거스름돈을 줘야한다.
손님에게 거슬러 줘야할 돈이 N원일 때, **거스름돈으로 주는 동전의 최소 개수**를 구하라!
단, N원은 항상 10의 배수이고, 동전은 500원, 100원, 50원, 10원이 있다.

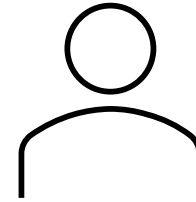
아이디어 : 큰 동전부터 거슬러주면 된다!



N = 1580원인 경우를 생각해보자.



직원
1580원

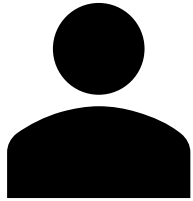


손님
0원

동전 수
0개

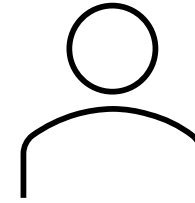
동전 수 : $1580 // 500 = 3$
남은 돈 : $1580 \% 500 = 80$

N = 1580원인 경우를 생각해보자.



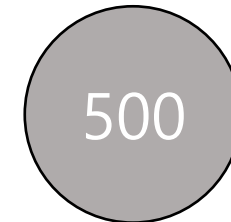
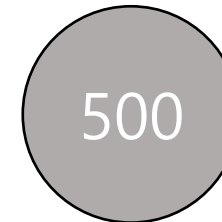
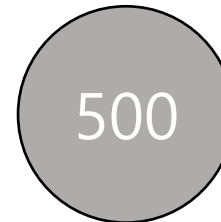
직원
80원

동전 수 : $80 // 50 = 1$
남은 돈 : $80 \% 50 = 30$

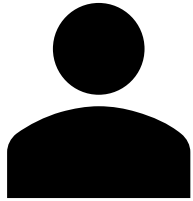


손님
1500원

동전 수
3개

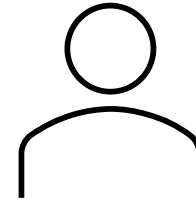


N = 1580원인 경우를 생각해보자.



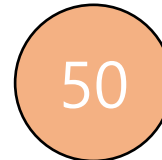
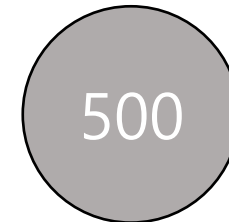
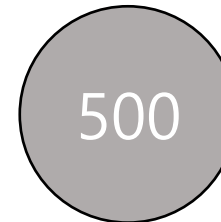
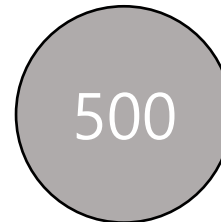
직원
30원

동전 수 : $30 // 10 = 3$
남은 돈 : $30 \% 10 = 0$

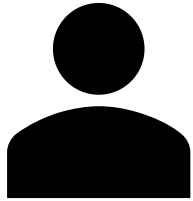


손님
1550원

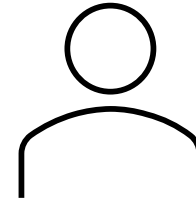
동전 수
4개



N = 1580원인 경우를 생각해보자.

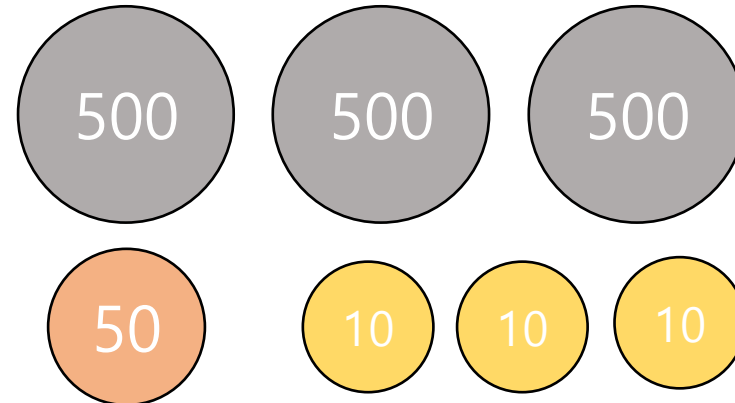


직원
0원



손님
1580원

동전 수
7개



코드로 작성

```
n = 1580
cnt = 0

coins = [500, 100, 50, 10]
# 가장 큰 단위부터

for coin in coins:
    cnt = cnt + n // coin
    n = n % coin

print(cnt)
```

이게 왜 가능했냐?

동전의 큰 단위가 항상 작은 단위의 배수 형태라서,
가장 큰 단위부터 거슬러 주면
거슬러 주는 동전의 개수가 최소가 되기 때문이다!

거스름돈 문제에서 그리디 알고리즘 적용하면 안 되는 경우도 있다.

Ex. N = 600인데 동전이 400, 300, 100 인 경우

그리디 → 400, 100, 100 (3개)

정답 → 300, 300 (2개)

동전 단위가 서로 배수 형태가 아니라면, DP로 풀어야한다
그리디가 매 단계에서 최적의 경우만 선택하는 것이라면
DP는 모든 경우를 저장해 놓고 계산해서 선택한다.

1로 만들기

정수 N 에 사용될 수 있는 연산은 다음과 같이 2가지이다.

1. N 이 K 로 나누어지면, N 을 K 로 나눈다.
2. N 에서 1을 뺀다.

정수 N 과 K 가 주어졌을 때, 위와 같은 연산 2개를 적절히 사용해서 1을 만들려고 한다.

연산을 사용하는 횟수의 최솟값을 출력하시오.

입력 조건 : 첫째 줄에 자연수 N, K ($2 \leq N, K \leq 100,000$) 가 공백으로 구분되어 주어진다.
이때, N 은 항상 K 보다 크거나 같다.

1로 만들기

정수 N 에 사용될 수 있는 연산은 다음과 같이 2가지이다.

1. N 이 K 로 나누어지면, N 을 K 로 나눈다.
2. N 에서 1을 뺀다.

정수 N 과 K 가 주어졌을 때, 위와 같은 연산 2개를 적절히 사용해서 1을 만들려고 한다.
연산을 사용하는 횟수의 최솟값을 출력하시오.

입력 조건 : 첫째 줄에 자연수 N, K ($2 \leq N, K \leq 100,000$) 가 공백으로 구분되어 주어진다.
이때, N 은 항상 K 보다 크거나 같다.

아이디어 : 1번과 2번 중에 N 이 더 작아지는 것을 선택하자
→ 최대한 많이 나누면 된다!



2 이상 자연수 K로 나누는 것이 N의 값을 빠르게 줄일 수 있다.

가정 : $N - 1 > N / K$

$$K > N / (N - 1)$$

$$K \geq 2, N \geq 2$$

$$2 * N > N$$

$$K \geq N / (N - 1)$$

1로 만들기

정수 N 에 사용될 수 있는 연산은 다음과 같이 2가지이다.

1. N 이 K 로 나누어지면, N 을 K 로 나눈다.
2. N 에서 1을 뺀다.

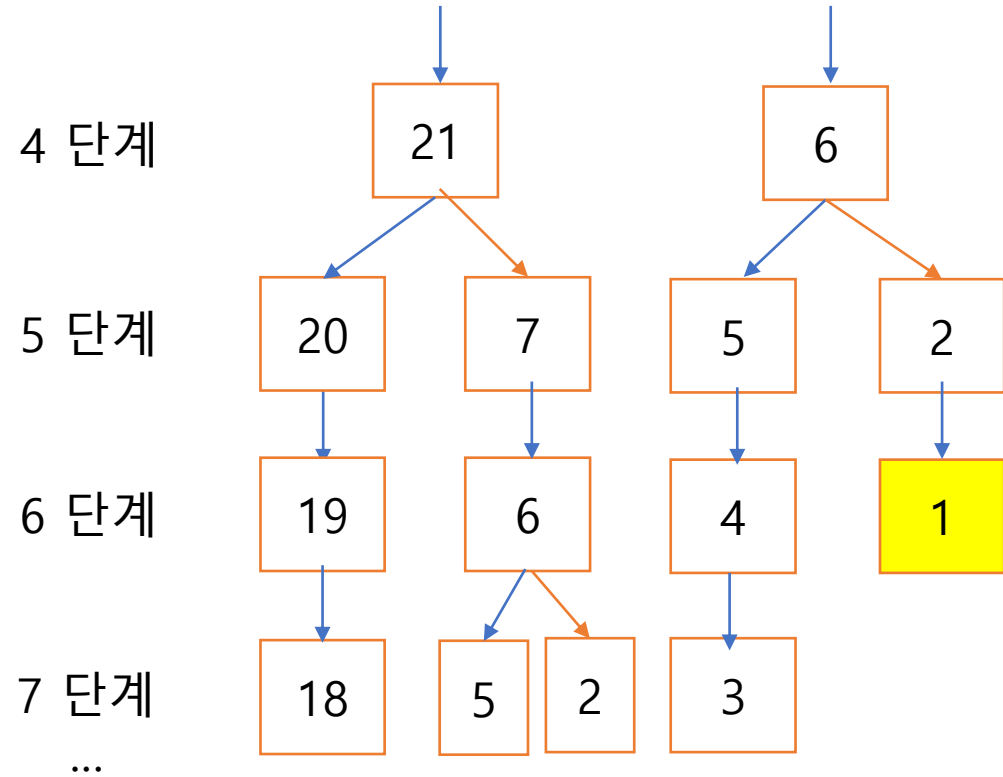
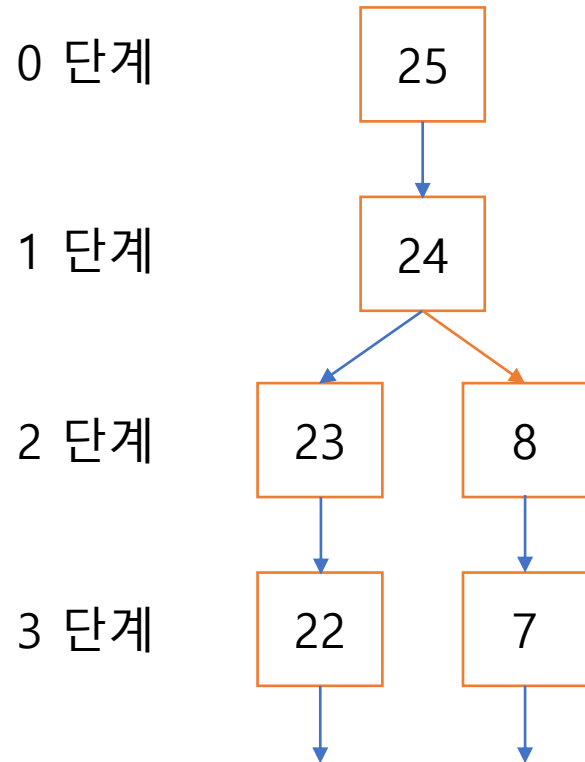
정수 N 과 K 가 주어졌을 때, 위와 같은 연산 2개를 적절히 사용해서 1을 만들려고 한다.
연산을 사용하는 횟수의 최솟값을 출력하시오.

입력 조건 : 첫째 줄에 자연수 N, K ($2 \leq N, K \leq 100,000$) 가 공백으로 구분되어 주어진다.
이때, N 은 항상 K 보다 크거나 같다.

아이디어 : 1번과 2번 중에 N 이 더 작아지는 것을 선택하자
→ 최대한 많이 나누면 된다!



N = 25, K = 3 인 경우



코드로 작성

```
n, k = map(int, input().split())

cnt = 0

while n >= k:
    if n % k == 0:
        n = n // k # n/k 로 하면 실수형이 저장된다.
        cnt += 1
    n -= 1
    cnt += 1

while n > 1:
    n -= 1
    cnt += 1

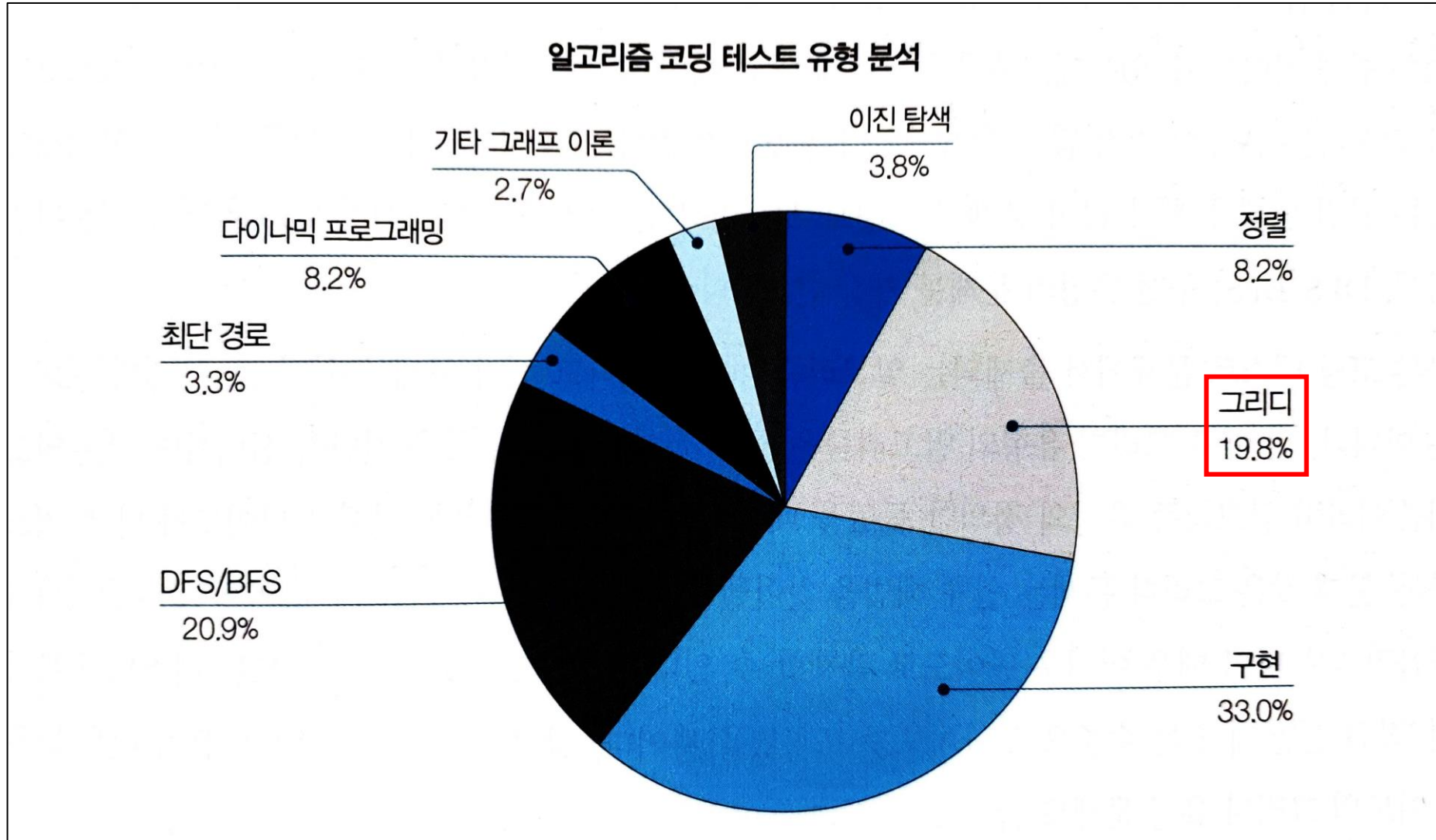
print(cnt)
```

이 코드는 개선 가능하다!
매 단계마다 1을 빼서
N이 K의 배수가 되도록 만드는 게 아니라,

매 단계에서 특정 값을 빼서
N이 K의 배수가 되도록 만들 수 있다.

한 번 생각해 보자 😊

추가로, 연산이 3가지 일 경우
1로 만드는 문제도 풀어보자!
(문제집에 넣어 났음)



출처 : 이것이 취업을 위한 코딩 테스트다 with 파이썬

문제 출제 범위가 매~~~우 넓다!



백준으로 파이썬 맛보기

- 8주차 문제집

문제 번호	제목
5585	2 거스름돈
11047	2 동전 0
11399	3 ATM
1946	1 신입 사원
1969	5 DNA
10610	5 30
1541	2 잃어버린 괄호
2875	3 대회 or 인턴
1783	4 병든 나이트
1931	2 회의실 배정
1744	4 수 묶기

회의실 꼭 풀어 보기! (자주 나옴)

