

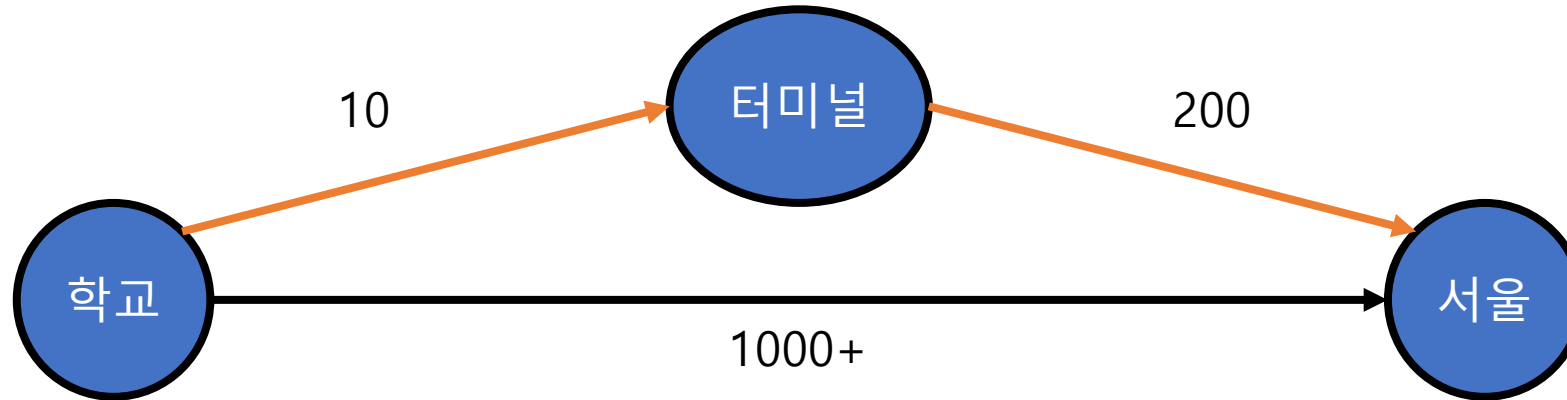
대경 HuStar아카데미 알고리즘 실습

그리디 기법

Greedy Method?

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

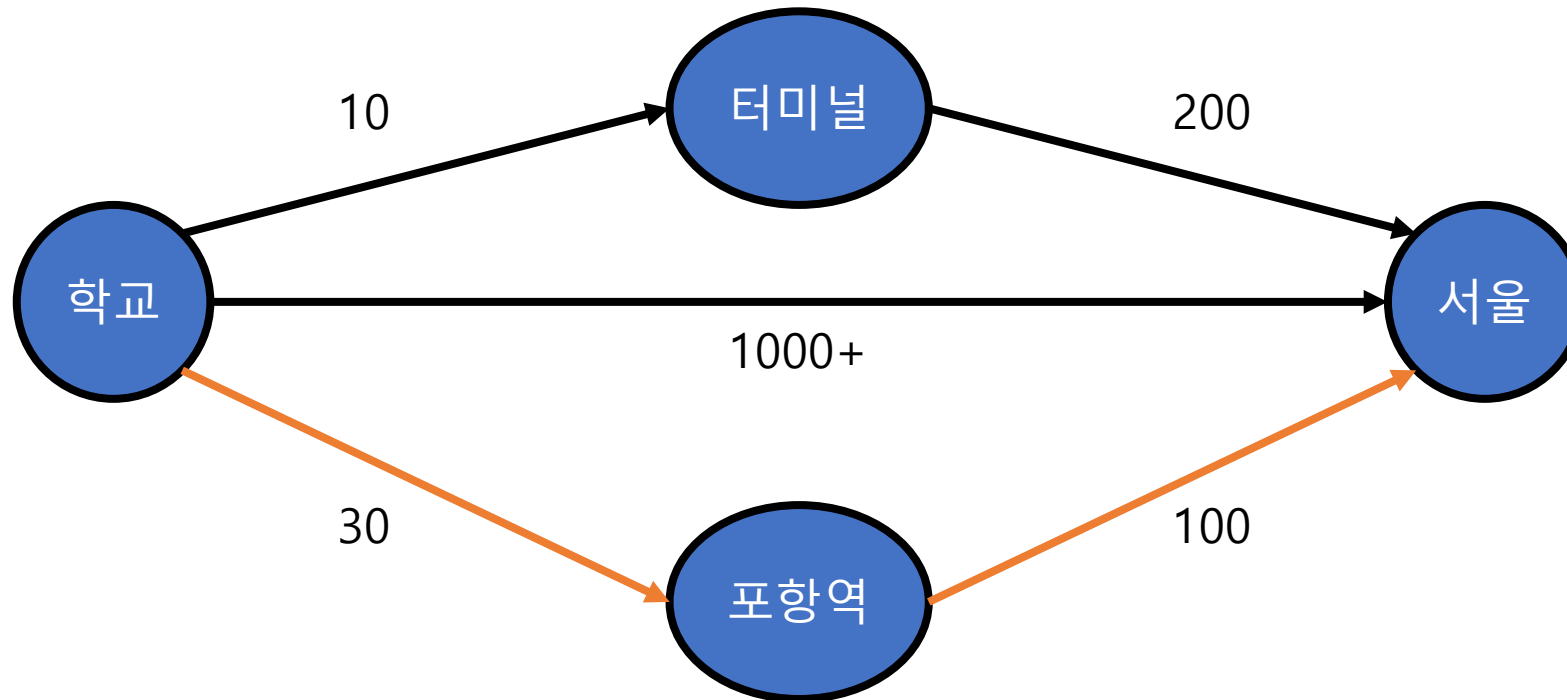
- 특정 상황에서 최적의 값을 찾는다.
- 알고리즘이 단순하고 빠르기 때문에 근사 알고리즘을 설계할 때 자주 채택한다.



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

```
t = int(input())
for _ in range(t):
    n = int(input())

    coinnum = 0
    for coin in coins:
        coinnum += n//coin
        n %= coin
    print(coinnum)
```

01. 세금 징수

주의!!
항상 그리디한 접근이 올바른 해답을
도출하는지 확인해 봐야한다.

Ex1: 사용 가능한 동전 [100,500]

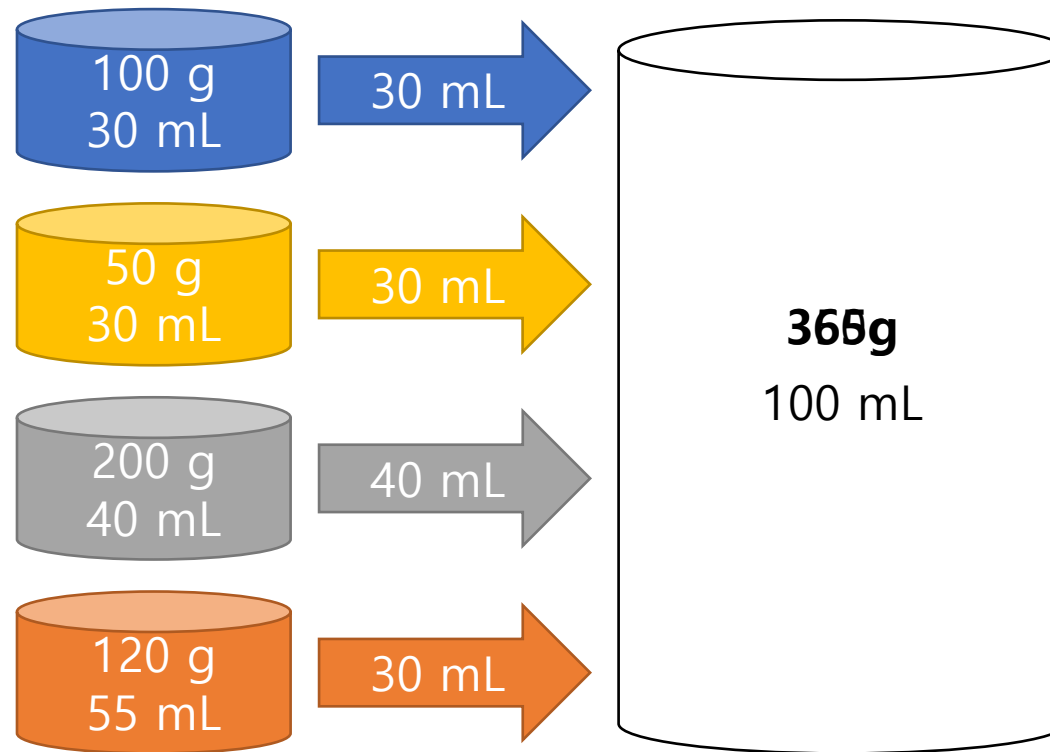
800원을 지불하는 방법 $\rightarrow 800 = 500 + 100 \times 3$ (4개)

Ex2: 사용 가능한 동전 [400,500]

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

02. 무거운 용액

여러 용액의 총 부피와 총 무게가 주어졌을 때, 주어진 용액을 합성하여 특정 부피의 용액을 만들 때 합성 용액의 최대 무게를 계산합니다.



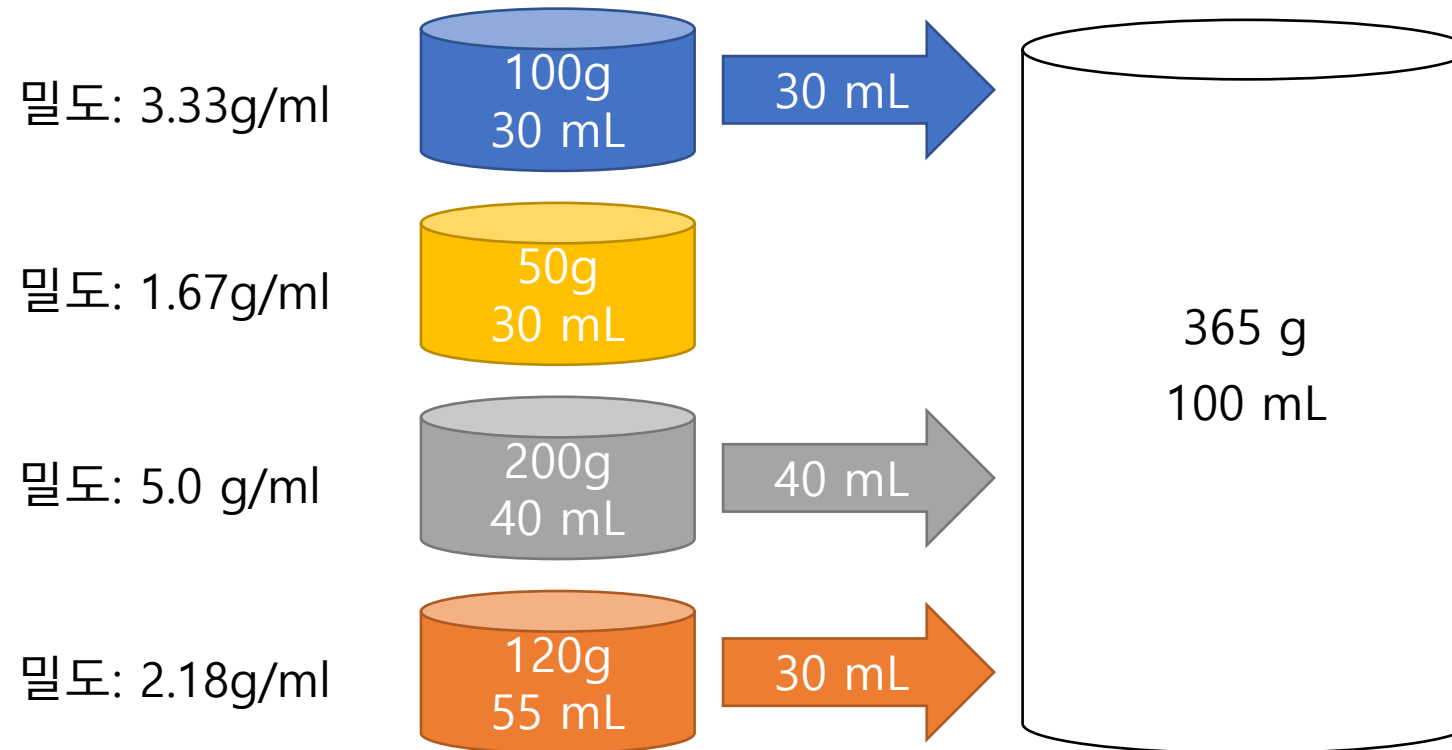
02. 무거운 용액

무거운 용액을 만들 때 가장 좋은 용액은 무엇인가요?

→ 보다 작은 공간을 차지하면서, 보다 더 무거운 용액

→ 같은 공간을 차지할 때 더 무거운 용액

→ **부피당 질량**이 무거운 용액



02. 무거운 용액

1. 질량 / 부피 로 내림차순 정렬
2. 질량 / 부피가 큰 용액부터 차례대로 넣는다.
→ 그 용액을 온전히 다 넣을 수 있다면?
→ 그 용액을 다 못 넣는다면?

밀도: 3.33g/ml



40 mL (200 g)

밀도: 1.67g/ml



30 mL (100 g)

밀도: 5.0 g/ml



30 mL ($30 \times 120 / 55$ g)

밀도: 2.18g/ml



100 mL

02. 무거운 용액

```
1  T = int(input())
2
3  for _ in range(T):
4      N,C = map(int,input().split())
5      liquidlist = []
6      for i in range(N):
7          w,v = map(int,input().split())
8          liquidlist.append((w/v,w,v))
9      liquidlist.sort(reverse=True)
10     maxg=0
11     for i in range(N):
12         if C>=liquidlist[i][2]:
13             maxg += liquidlist[i][1]
14             C-=liquidlist[i][2]
15         else:
16             maxg+=C*(liquidlist[i][0])
17             break
18     print(int(maxg))
```

Project01. 부산에서 서울로

부산에서 서울까지 갈 때 최소 횟수로 주유소에서 기름 채우기

*Hint

- 이미 지난 주유소는 다시 갈 필요가 없다. (선형 탐색)
- 서울을 기름 안 넣는 주유소처럼 취급해보자!

Project01. 부산에서 서울로

Idea: 갈 수 있는 가장 먼 주유소를 가자

유지하는 변수: 지금까지 주유한 횟수, 가장 마지막에 주유한 주유소

Project01. 부산에서 서울로

기름 용량: 20



부산: 0

7

14

19

29

36

51

71

81

서울: 100



주유 횟수: 0

마지막 주유소: 0

Project01. 부산에서 서울로

기름 용량: 20



부산: 0

7

14

19

29

36

51

71

81

서울: 100



주유 횟수: 0

마지막 주유소: 0

Project01. 부산에서 서울로

기름 용량: 20



부산: 0 7 14 19 29 36 51 71 81 서울: 100



주유 횟수: 0

마지막 주유소: 0

Project01. 부산에서 서울로

기름 용량: 20



부산: 0 7 14 19 29 36 51 71 81 서울: 100



주유 횟수: 0

마지막 주유소: 0

Project01. 부산에서 서울로

기름 용량: 20



부산: 0 7 14 19 29 36 51 71 81 서울: 100



주유 횟수: 1

마지막 주유소: 19

Project01. 부산에서 서울로

기름 용량: 20



부산: 0 7 14 19 29 36 51 71 81 서울: 100



주유 횟수: 1

마지막 주유소: 19

Project01. 부산에서 서울로

기름 용량: 20



부산: 0 7 14 19 29 36 51 71 81 서울: 100



주유 횟수: 2

마지막 주유소: 36

Project01. 부산에서 서울로

기름 용량: 20



부산: 0 7 14 19 29 36 51 71 81 서울: 100



주유 횟수: 3

마지막 주유소: 51

Project01. 부산에서 서울로

기름 용량: 20



부산: 0 7 14 19 29 36 51 71 81 서울: 100

주유 횟수: 4

마지막 주유소: 71

Project01. 부산에서 서울로

기름 용량: 20



부산: 0 7 14 19 29 36 51 71 81 서울: 100

주유 횟수: 5

마지막 주유소: 81

Project01. 부산에서 서울로

생각해봐야할 문제

- 부산에서 서울까지 주유소를 한 번도 거치지 않고 갈 수도 있다.
- 마지막 주유소까지 도달했으나, 서울까지 가지 못할 수 있다.

여러 가지 상황을 잘 고려하면서 해결해야 하는 문제!