

김범수



# 그리디

**Greedy**

By POSCAT



## Contents

---

그리디란?	03
-------	----

---

동전 0	05
------	----

---

회의실 배정	08
--------	----

---

---

---

# 그리디란?

- 그리디(greedy)는 탐욕 알고리즘이라고도 불리는 알고리즘으로, 현재 상태에서 가장 최선의 선택을 하는 기법입니다.
- 저번에 DP로 풀었던 1로 만들기 문제를 생각해 봅시다.  $x$ 라는 숫자가 주어지면 1을 빼기, 2로 나누기, 3으로 나누기 이 세 연산들을 적용해서 최소한의 연산으로 1로 만드는 문제입니다.
- 이 때 우리는 세 연산 중 어떤 연산을 사용하는 게 가장 좋은 지 모르기 때문에(상황에 따라 가장 좋은 연산이 달라지기 때문에) DP를 사용해서 풀었습니다. 따라서  $DP[x-1]$ ,  $DP[x/2]$ ,  $DP[x/3]$  중 가장 작은 값을 선택하는 방식으로 풀었습니다.
- 하지만 세 연산이 1을 빼기, 2를 빼기, 3을 빼기 이렇게 바뀌면 어떻게 될까요? 그렇다면 굳이 DP를 쓸 필요 없이  $x$ 가 3 이상이면 무조건 3을 빼기 연산을 적용하는 것이 가장 좋은 연산이 됩니다.
- 이처럼 기존 1로 만들기 문제는  $/3$  연산을 적용하면  $x$ 가 매우 작아져 그 순간에는 가장 좋아보지만, 결과적으로는 가장 좋은 것이 아닌 결과가 나올 수도 있습니다. 하지만 우리가 새로 만든 1로 만들기 문제에서는  $-3$ 을 선택하는 것이 그 순간에 가장 좋으면서, 결과적으로도 가장 좋은 결과를 냅니다.

# 그리디란?



- 그리디 알고리즘을 적용하기 위해서는 “현재 상태에서 가장 최선의 선택을 하는 것이 결과적으로 가장 좋은 결과를 내는가?”를 먼저 증명해야 합니다.
- 증명은 보통 귀류법을 사용해서 모순을 이끌어내는 방법으로 진행됩니다. 즉 그리디를 사용한 정답보다 더 좋은 정답이 존재한다고 가정을 하고, 여기서 모순을 이끌어냅니다.
- 그리디는 알고리즘 자체는 간단하지만, 문제가 그리디란 것을 파악하는 것, 그리고 그리디라는 것을 증명하는 것 자체가 어렵기 때문에 문제를 풀어보면서 그리디를 익혀보겠습니다.

# 동전 0



## 문제

준규가 가지고 있는 동전은 총  $N$ 종류이고, 각각의 동전을 매우 많이 가지고 있다.

동전을 적절히 사용해서 그 가치의 합을  $K$ 로 만들려고 한다. 이때 필요한 동전 개수의 최솟값을 구하는 프로그램을 작성하시오.

## 입력

첫째 줄에  $N$ 과  $K$ 가 주어진다. ( $1 \leq N \leq 10, 1 \leq K \leq 100,000,000$ )

둘째 줄부터  $N$ 개의 줄에 동전의 가치  $A_i$ 가 오름차순으로 주어진다. ( $1 \leq A_i \leq 1,000,000, A_1 = 1, i \geq 2$ 인 경우에  $A_i$ 는  $A_{i-1}$ 의 배수)

## 출력

첫째 줄에  $K$ 원을 만드는데 필요한 동전 개수의 최솟값을 출력한다.

## Example Input



10 4200

1

5

10

50

100

500

1000

5000

10000

50000

## Example Output



6

// 1000 \* 4 + 100 \* 2

## 동전 0



- 거스름돈을 줄 때, 100원을 주고 10원을 주는 거나 10원을 주고 100원을 주나 거스름돈을 주는 순서는 결과에 영향이 없으므로, 여기서는 가장 금액이 큰 동전부터 작은 동전 순서로 거스름돈을 줍니다.
- 그리디하게 생각해보면 현재 내가 줄 수 있는 동전 중 가장 큰 단위금액의 동전을 주는 것이 가장 이득이라 생각할 수 있습니다.
- 예를 들어 예제 입력처럼 4200원이 남았다면, 줄 수 있는 동전 중 가장 큰 단위인 1000원을 주는 것이 가장 이득입니다.

# 회의실 배정



- 다음의 문제를 생각해봅시다.

## 문제

한 개의 회의실이 있는데 이를 사용하고자 하는  $N$ 개의 회의에 대하여 회의실 사용표를 만들려고 한다. 각 회의  $i$ 에 대해 시작시간과 끝나는 시간이 주어져 있고, 각 회의가 겹치지 않게 하면서 회의실을 사용할 수 있는 회의의 최대 개수를 찾아보자. 단, 회의는 한번 시작하면 중간에 중단될 수 없으며 한 회의가 끝나는 것과 동시에 다음 회의가 시작될 수 있다. 회의의 시작시간과 끝나는 시간이 같을 수도 있다. 이 경우에는 시작하자마자 끝나는 것으로 생각하면 된다.

- 만약 최대한 많은 회의실을 사용하고 싶다면, 무엇을 기준으로 회의를 선택해야 할까요?
- 다음 회의는 이전 회의가 끝난 이후에 시작할 수 있으므로, 회의가 일찍 끝날수록 우리가 선택할 수 있는 회의의 폭이 넓어질 것입니다.
- 따라서 빨리 끝나는 회의를 선택하는 것이 좋아보입니다.



# 회의실 배정



- 실제로 현재 선택할 수 있는 회의 중 가장 빨리 끝나는 회의를 선택하는 것이 이 문제의 풀이입니다. 아래의 예제 입력을 한번 풀어봅시다.
- 먼저 가장 빨리 끝나는 회의는 (1, 4)입니다. 먼저 (1, 4) 회의를 선택합니다.
- 4 이후에 시작하는 회의 중 가장 빨리 끝나는 회의는 (5, 7)입니다. 따라서 (5, 7) 회의를 선택합니다.
- 7 이후에 시작하는 회의 중 가장 빨리 끝나는 회의는 (8, 11)입니다. 따라서 (8, 11)을 선택합니다.
- 마찬가지로 (12, 14) 회의를 선택합니다.
- 이제 더 이상 선택할 회의가 존재하지 않으므로 종료합니다.
- 즉 최대 4개의 회의를 진행할 수 있습니다.

예제 입력 1 복사

```
11
1 4
3 5
0 6
5 7
3 8
5 9
6 10
8 11
8 12
2 13
12 14
```

## 회의실 배정

- 이제 이 과정이 정당한 지를 증명해 봅시다. 귀류법을 통해, 그리디를 통해서 최대  $n$ 개의 회의를 개최할 수 있지만  $n + 1$ 회 이상의 회의를 진행하는 방법이 존재한다고 가정해 봅시다.
- 그리디로 선택한 회의는  $(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)$ 이 될 것이고, 이보다 더 많이 회의를 진행하는 방법  $(x'_1, y'_1), (x'_2, y'_2), \dots (x'_{n'}, y'_{n'})$  이라 합니다. ( $n < n'$ )
- 이 때 두 회의 스케줄 중 처음으로 다른 회의가 존재하는 시점을  $i$ 라 합니다. 즉  $(x_i, y_i) \neq (x'_i, y'_i)$ 를 만족하는 가장 작은  $i$ 를 의미합니다. 이 때, 우리는 그리디로 선택했으므로,  $y_i \leq y'_i$ 를 만족합니다.
- 하지만 그리디 과정을 거치면  $(x_1, y_1), (x_2, y_2), \dots (x_i, y_i), (x'_{i+1}, y'_{i+1}) \dots (x'_{n'}, y'_{n'})$ 의 스케줄, 또는 그보다 많은 회의를 포함하는 스케줄을 만들 수 있습니다. ( $y_i \leq y'_i$ 이기 때문) 따라서  $n < n'$ 임에 모순이 발생합니다.
- 결론적으로, 그리디를 통해 개최한 회의 수보다 더 많은 수의 회의를 개최하는 것은 불가능합니다.

## 회의실 배정



- 하지만  $n$ 범위가 최대 10만이기 때문에, 회의를 선택할 때마다 모든 회의를 보고 그 중 가장 빨리 끝나는 회의를 선택하는 방식으로 구현하면  $O(n^2)$ 의 시간이 걸리기 때문에 시간 초과가 발생합니다.
- 따라서 주어진 회의를 회의가 끝나는 시간을 기준으로 정렬을 먼저 진행합니다. 그 후 가장 빨리 끝나는 회의를 선택할 때 순차적으로 빨리 끝나는 회의부터 보면서 선택을 합니다.
- 이 방식을 따르면 정렬 시간을 제외하고  $O(n)$ 의 시간에 문제를 풀 수 있습니다.
- 정렬 시간까지 포함하면  $O(n \log n)$ 안에 문제를 풀 수 있습니다.
- 직접 예제를 풀어보면서 과정을 이해해 봅시다.


# 회의실 배정

...

- 먼저 변수  $t$ 에 가장 최근에 끝난 회의 시간을 저장하고, 변수  $cnt$ 에 진행한 회의의 개수를 저장합니다.
- 회의 시간의 범위는 0 이상  $2^{32}$  이하이므로,  $t = 0$ ,  $cnt = 0$ 으로 초기화합니다.
- 파란색 화살표로 현재 보고 있는 회의가 무엇인 지 표시합니다.

$t = 0$

$cnt = 0$




11
1 4
3 5
0 6
5 7
3 8
5 9
6 10
8 11
8 12
2 13
12 14

## 회의실 배정

- 먼저  $t = 0$ 이고, 현재 보고 있는 회의는 그 이후인 1에 시작하므로, 지금 검사하고 있는 회의를 진행합니다.
- $t = 4$ 로 갱신하고 개수를 1만큼 증가시킵니다.

$t = 4$

$\text{cnt} = 1$




```
11
1 4
3 5
0 6
5 7
3 8
5 9
6 10
8 11
8 12
2 13
12 14
```

## 회의실 배정

- ...
- 지금 보고 있는 회의는 가장 마지막에 진행한 회의가 끝나는 시점인 4보다 더 빠른 시점인 3에 시작하므로 현재 회의는 진행할 수 없습니다. 따라서 건너뜁니다.

t = 4

cnt = 1




```
11
1 4
3 5
0 6
5 7
3 8
5 9
6 10
8 11
8 12
2 13
12 14
```

## 회의실 배정

- ...
- 지금 보고 있는 회의는 가장 마지막에 진행한 회의가 끝나는 시점인 4보다 더 빠른 시점인 0에 시작하므로 현재 회의는 진행할 수 없습니다. 따라서 건너뛰니다.

t = 4

cnt = 1



11	
1	4
3	5
0	6
5	7
3	8
5	9
6	10
8	11
8	12
2	13
12	14

## 회의실 배정

...

- $t = 4$ 이고, 현재 보고 있는 회의는 그 이후인 5에 시작하므로, 지금 검사하고 있는 회의를 진행합니다.
- $t = 7$ 로 갱신하고 개수를 1만큼 증가시킵니다.

$t = 7$

cnt = 2

11  
1 4  
3 5  
0 6  
5 7  
3 8  
5 9  
6 10  
8 11  
8 12  
2 13  
12 14





## 회의실 배정

- ...
- 지금 보고 있는 회의는 가장 마지막에 진행한 회의가 끝나는 시점인 7보다 더 빠른 시점인 3에 시작하므로 현재 회의는 진행할 수 없습니다. 따라서 건너뜁니다.

t = 7

cnt = 2

11  
1 4  
3 5  
0 6  
5 7  
→ 3 8  
5 9  
6 10  
8 11  
8 12  
2 13  
12 14

## 회의실 배정

- ...
- 지금 보고 있는 회의는 가장 마지막에 진행한 회의가 끝나는 시점인 7보다 더 빠른 시점인 5에 시작하므로 현재 회의는 진행할 수 없습니다. 따라서 건너뛵니다.

t = 7

cnt = 2

11  
1 4  
3 5  
0 6  
5 7  
3 8  
→ 5 9  
6 10  
8 11  
8 12  
2 13  
12 14

## 회의실 배정

- ...
- 지금 보고 있는 회의는 가장 마지막에 진행한 회의가 끝나는 시점인 7보다 더 빠른 시점인 6에 시작하므로 현재 회의는 진행할 수 없습니다. 따라서 건너뜁니다.

t = 7

cnt = 2

11  
1 4  
3 5  
0 6  
5 7  
3 8  
5 9  
→ 6 10  
8 11  
8 12  
2 13  
12 14

## 회의실 배정

...

- $t = 7$ 이고, 현재 보고 있는 회의는 그 이후인 8에 시작하므로, 지금 검사하고 있는 회의를 진행합니다.
- $t = 11$ 로 갱신하고 개수를 1만큼 증가시킵니다.

$t = 11$

cnt = 3

11	
1	4
3	5
0	6
5	7
3	8
5	9
6	10
8	11
8	12
2	13
12	14



## 회의실 배정

...

- 지금 보고 있는 회의는 가장 마지막에 진행한 회의가 끝나는 시점인 11보다 더 빠른 시점인 8에 시작하므로 현재 회의는 진행할 수 없습니다. 따라서 건너뛴다.

t = 11

cnt = 3

```
11
1 4
3 5
0 6
5 7
3 8
5 9
6 10
8 11
8 12
2 13
12 14
```



## 회의실 배정

...

- 지금 보고 있는 회의는 가장 마지막에 진행한 회의가 끝나는 시점인 11보다 더 빠른 시점인 2에 시작하므로 현재 회의는 진행할 수 없습니다. 따라서 건너뜁니다.

t = 11

cnt = 3

```
11
1 4
3 5
0 6
5 7
3 8
5 9
6 10
8 11
8 12
2 13
12 14
```



## 회의실 배정

...

- $t = 11$ 이고, 현재 보고 있는 회의는 그 이후인 12에 시작하므로, 지금 검사하고 있는 회의를 진행합니다.
- $t = 14$ 로 갱신하고 개수를 1만큼 증가시킵니다.
- 모든 회의에 대해 검사했으니, 진행할 수 있는 회의의 최대 개수는 4임을 알 수 있습니다.

$t = 14$

$\text{cnt} = 4$

11	
1	4
3	5
0	6
5	7
3	8
5	9
6	10
8	11
8	12
2	13
12	14



# 연습 문제

• • •

- <https://www.acmicpc.net/problem/11047> 동전 0
- <https://www.acmicpc.net/problem/11399> ATM, 간단한 그리디
- <https://www.acmicpc.net/problem/2217> 로프, 간단한 그리디
- <https://www.acmicpc.net/problem/1931> 회의실 배정
- <https://www.acmicpc.net/problem/15903> 카드 합체 놀이, 어떻게 합체를 해야 카드에 쓰인 숫자의 합이 작아질 지를 고민해보자.
- <https://www.acmicpc.net/problem/1700> 멀티탭 스케줄링, 오랫동안 쓰지 않을 코드를 빼놓자.