

# 5주차 문제 풀이

모든 코드는 [www.github.com/jjwdi0/](https://www.github.com/jjwdi0/) 에 있습니다~

# 문제 목록

- ATM
- 회의실배정
- 신입 사원
- 저울
- 택배
- 컵라면
- Minimum Scalar Product (Large)
- 공주님의 정원

# 이론적 배경 - 그리디 알고리즘

- '욕심쟁이 기법' 이라고도 불림.
- 예시를 들어 보자.
- 만약 당신 앞에 100만원이 있고, 가장 많은 돈을 가지고 싶을 때, 얼마나 가져올 것인가?
- 답은 '전부 다' 이겠지??
- 이렇게 '욕심 내어' 선택한 결과가 전체 문제에서도 최적을 보장한다는 것이 그리디 알고리즘이다.

# 이론적 배경 – 그리디 알고리즘

- 물론 모든 문제에 이러한 방법이 성립하는 것은 아니다. (그러면 세상이 참 편할 것이다)
- 문제에 따라 직관적으로 방법이 떠오를 수도 있고, 복잡한 수학적 지식으로 유도할 수도 있다.
- 그래서 매우 감을 잡기 어렵다.
- 대표적인 몇 가지 유형을 풀어 보자.
- +추가) Matroid에 대해 찾아보자. (시간이 되면)

# 1. ATM

- 그냥 직관적으로도 인출에 걸리는 시간이 적은 사람부터 인출해야 최적임을 알 수 있다.
- 정렬하면 된다. 시간복잡도는  $O(n \log n)$ .

# 1. ATM

- 간단한 증명을 하도록 하겠다. 참고로 대부분의 그리디 문제는 간단한 증명을 통해 정당성을 확보한 다음 손을 대는 것이 좋다. (그려야 삽질 안함.)
- $P_i$ 를 무작위로 배열하고,  $P_i > P_j$ 이고  $i < j$ 인 쌍을 바꿀수록 최적해가 됨을 증명하자.
- $P_i$ 는 전체 해에서  $n - i + 1$ 번 더해지므로 이는  $P_i * (n - i + 1) + P_j * (n - j + 1)$  과  $P_j * (n - i + 1) + P_i * (n - j + 1)$ 의 크기를 비교하는 것과 같다.
- $n - i + 1 > n - j + 1$ 이므로 앞의 식이 더 크다는 것을 알 수 있다. (이는 재배열 부등식 또는 벡터의 내적을 이용해 증명 가능하다.)
- 따라서  $P_i$ 를 오름차순으로 정렬해야 최선이다.

## 2. 회의실 배정

- 다른 말로 설명하자면 "선분 그래프에서 최대 독립 집합의 크기는 얼마인가?"
- 이는 간단한 그리디 알고리즘으로 해결 가능하다.  $O(n \log n)$ .
- 대표적인 예시로 구간 스케줄링 문제 라고도 하며 매우 많은 방법으로 응용될 수 있다. (APIO 2009 – 컨벤션 센터)
- 구간을  $[s, e]$ 로 표시할 때,  $e$ 가 같으면  $s$ 가 작은 것이, 아니면  $e$ 가 작은 것이 앞에 오게 정렬한다.
- 그 다음 앞에서부터 추가 가능하면 현재의 해에 추가한다.
- 구한 해가 답이다.

## 2. 회의실 배정

- 증명:
- 먼저 어떤 최적해 집합  $S = \{A_1, A_2, \dots, A_k\}$ 가 있다고 하자.
- 이 때  $A_1$ 을 더 빨리 끝나는 회의  $A_1'$ 으로 바꿔도 여전히 최적해가 된다.
- 이러한 방법으로 구한  $S' = \{A_1', A_2', \dots, A_k'\}$  역시 최적해이다.
- 증명 끝.



### 3. 신입 사원

- 그리디 알고리즘이라고 하기에는 애매하지만...
- 먼저 한쪽 순위를 기준으로 정렬하여, 나머지 순위를 스택을 이용하여 스택 내부를 내림차순으로 유지시켜가면서 해를 구성하면 된다.
- 시간복잡도  $O(n \log n)$ .
- 이와 같은 기법은 다른 문제에도 흔히 쓰인다. 예 : [USACO 2008 Gold – Acquire](#)

## 4. 저울

- $[1, x]$  구간까지 모든 무게를 만들 수 있을 때, 여기에 새로운  $x'$ 의 무게를 가진 추가 추가된다고 하자.
- 그렇다면  $[1, x] + [x', x + x']$ 의 구간을 만들 수 있을 것이다.
- 이 때 구간이 두 개라면?
- 첫 번째 구간 바로 오른쪽 수가 답이다.
- 시간복잡도  $O(n \log n)$ . (정렬)

## 5. 택배

- 앞에서 다루었던 구간 스케줄링 문제.
- 마을을 시간, 박스 개수를 회의 숫자, 보내는 마을 / 받는 마을을 회의라 생각하면 똑같이 하면 된다.
- 이를 응용한 문제가 [USACO 2009 – Fair Shuttle](#)

## 6. 컵라면

- 이 문제는 풀이가 여러가지이다.
- 내 풀이는 정렬 + Union-Find 트리 인데, 나처럼 푼 사람은 한 명도 못봤으니... 권장하지는 않는다.
- 내 풀이:
- 먼저 컵라면 수에 대해 내림차순으로 정렬하고, 각 문제를 데드라인에서 가장 가까운 날짜에 배치한다.
- 하루에 한 문제이므로 겹치지 않게 Union-Find 트리를 이용해 관리할 수 있다.
- 정렬에  $O(n \log n)$ , 문제 배정에  $O(n)$ 이므로 총 시간복잡도  $O(n \log n)$ .

## 7. Minimum Scalar Product

- 노란 책에 나와있는 문제.
- 수학 좀 잘한다 싶으면 재배열 부등식을 이용하면 되고, 아니면 직감적으로 큰 것은 작은 것과, 작은 것은 큰 것과 곱한 것이 가장 작다는 것을 알 것이다.
- 즉 하나는 오름차순, 하나는 내림차순으로 정렬하여 곱하면 최소다.
- 시간복잡도  $O(n \log n)$ .

## 8. 공주님의 정원

- 구간 스케줄링 문제...
- 설명은 생략한다.
- 구현에 좀 더 도움이 되는 정보를 주자면, 날짜가 12월 31일까지라는 것을 이용하자.

- 다음주는 자료구조(?) 예정이다 ㅎㅎ