

# Maximum Number of Weeks for Which You Can Work

- Greedy -

안태진([taejin7824@gmail.com](mailto:taejin7824@gmail.com))

GitHub([github.com/taejin1221](https://github.com/taejin1221))

상명대학교 소프트웨어학과

201821002

# Contents

---

- Problem
- Approach
- Code

# Problem

- 문제 (1/2)
  - 프로젝트가 있고, 각 프로젝트당 몇일을 해야하는지 주어짐
  - 하루에 하나의 프로젝트만 수행 가능
  - 전날에 했던 프로젝트는 오늘 할 수 없음
  - 최대 며칠동안 프로젝트를 진행할 수 있을까?

## 1953. Maximum Number of Weeks for Which You Can Work

Medium  242  78  Add to List  Share

There are  $n$  projects numbered from  $0$  to  $n - 1$ . You are given an integer array `milestones` where each `milestones[i]` denotes the number of milestones the  $i^{\text{th}}$  project has.

You can work on the projects following these two rules:

- Every week, you will finish **exactly one** milestone of **one** project. You **must** work every week.
- You **cannot** work on two milestones from the same project for two **consecutive** weeks.

Once all the milestones of all the projects are finished, or if the only milestones that you can work on will cause you to violate the above rules, you will **stop working**. Note that you may not be able to finish every project's milestones due to these constraints.

Return the **maximum** number of weeks you would be able to work on the projects without violating the rules mentioned above.

# Problem

---

- 문제 (2/2)

- 예제 1

**Input:** milestones = [1,2,3]

**Output:** 6

- 0번 2번 1번 2번 1번 2번 순으로 하면 6일동안 프로젝트 진행 가능

- 예제 2

**Input:** milestones = [5,2,1]

**Output:** 7

- 0번 1번 0번 1번 0번 2번 0번 순으로 하면 7일동안 프로젝트 진행 가능
    - 0번은 4일만 진행했기 때문에 마무리하지 못했지만,  
어디에 끼워 넣든 연속 2일이기 때문에 마무리 불가능

# Contents

---

- Problem
- Approach
- Code

# Approach

---

- Approach (1/3)
  - 최대로 만들기 위해선 가장 큰 프로젝트를 중간에 배치하여야 함
    - 다른 프로젝트들에서 한 프로젝트를 먼저 끝내야 할까? 번갈아가면서 끝내야할까?
      - 가장 큰 프로젝트를 먼저 끝내야할까? 가장 작은 프로젝트를 먼저 끝내야할까?
  - 여러가지 예시로 알아낸 것
    - 선형 증가면 무조건 가능함
      - e.g., [1, 2, 3, 4] -> 1, 2, 4, 2, 4, 3, 4, 3, 4, 3

# Approach

---

- Approach (2/3)

- 이때 떠오른 생각!

- 모든 프로젝트를 순서 상관없이 끝내긴 위해서  $sum(\text{남은 일수})$ 가 필요함
    - 그렇다면 가장 큰 프로젝트를 제외한 나머지 프로젝트를 일렬로 순서대로 배치하고, 가장 많이 남은 프로젝트를 그 사이 사이에 끼 넣으면 되지 않을까?
      - e.g., [1, 2, 3, 4] -> 1, 4, 2, 4, 3, 4, 3, 4, 3
      - [1, 2, 3, 4, 5] -> 1, 2, 5, 2, 3, 5, 3, 5, 3, 4, 5, 4, 5, 4, 4
      - 4옆을 제외한 어디든 재배치 가능
      - [1, 2, 3, 4, 5, 6] -> 1, 2, 6, 2, 3, 6, 3, 6, 3, 4, 6, 4, 6, 4, 6, 4, 5, 5, 5, 5, 5
      - 5는 중간에 무조건 끼 넣을 수 있음

# Approach

---

- Approach (3/3)
  - 즉, 가장 큰 프로젝트가 나머지 프로젝트의 합보다 작으면 무조건 가능
    - 안되는 경우 가장 큰 프로젝트가 나머지 프로젝트의 합보다 2이상 클 경우
      - e.g, A, A, A, A, ..., A, A일때 가장 큰 프로젝트가 중간에 들어가야 함
      - 따라서  $sum(\text{나머지 프로젝트}) + 1$ 이 최대 큰 프로젝트를 수행할 수 있는 날짜
  - $\therefore answer = S + \min(\max(milestones), S + 1)$ 
    - $S = sum(milestones) - \max(milestones)$
  - $sum(milestones)$ 와  $\max(m)$ 을 구하면 끝!



# Contents

---

- Problem
- Approach
- Code

# Code

- $sum(milestones)$ 와  $max(milestones)$  구하기

```
long long maxVal = 0ll, sum = 0ll;
for (int i = 0; i < milestones.size(); i++) {
    sum += milestones[i];
    maxVal = max(maxVal, (long long)milestones[i]);
}
```

milestones의 최대 값이 10억이고 n이 10만이기 때문에 long long으로 선언

- $S$ 와  $answer$  구하기

```
sum -= maxVal;

return sum + min(maxVal, sum + 1);
```

# Code

---

- Result

Success [Details >](#)

Runtime: **195 ms**, faster than **22.49%** of C++ online submissions for  
Maximum Number of Weeks for Which You Can Work.

Memory Usage: **75.6 MB**, less than **55.97%** of C++ online submissions for  
Maximum Number of Weeks for Which You Can Work.

---

감사합니다!

---