

ICPC N주차 문제 풀이

Maximum Number of Weeks for Which You Can V

소프트웨어학과 201921017 이지우

Step1 – 접근: 힌트보기

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^{th} project has.

You can work on the projects following these two rules:

동일한 프로젝트의 마일스톤을 연속으로 진행할 수 없음

- 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.

Hide Hint 1

가능한 가장 큰 마일스톤을 가진 프로젝트를 기준으로 진행

Work on the project with the largest number of milestones as long as it is possible.

Hide Hint 2

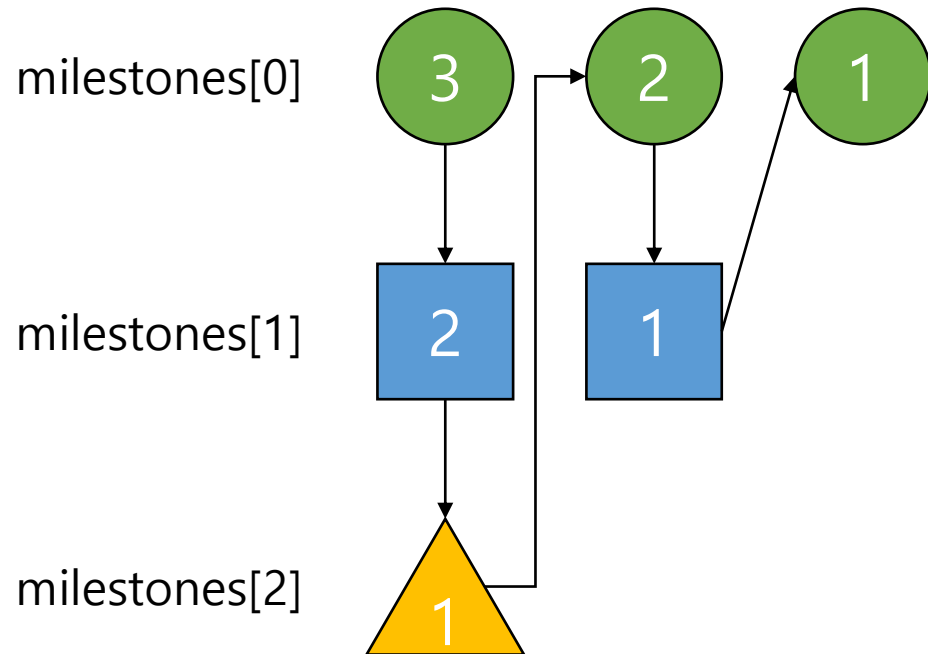
가장 큰 마일스톤을 가진 프로젝트가 week 수에 영향을 미치는가?

Does the project with the largest number of milestones affect the number of weeks?

Step1 – 접근

milestones = [1, 2, 3]

= [3, 2, 1]



→ 최대 6주

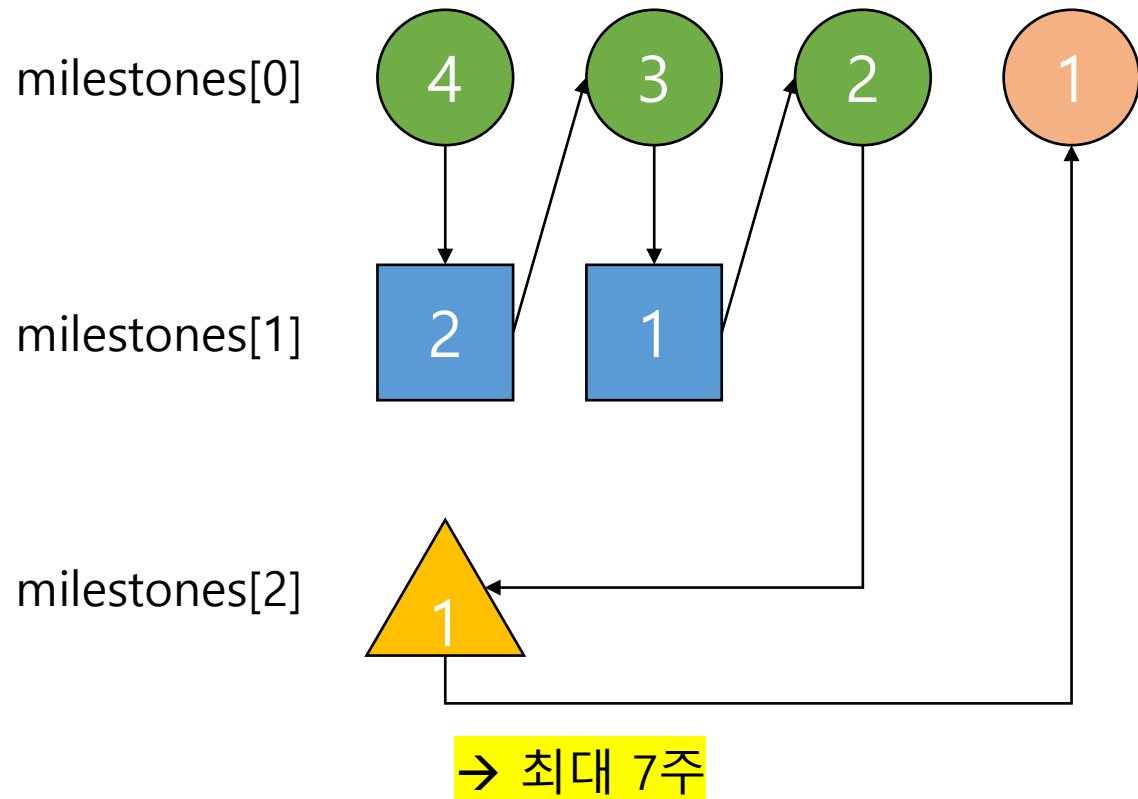
Hide Hint 2

Does the project with the largest number of milestones affect the number of weeks?

가장 큰 마일스톤을 가진 프로젝트가 week 수에 영향을 미치는가?

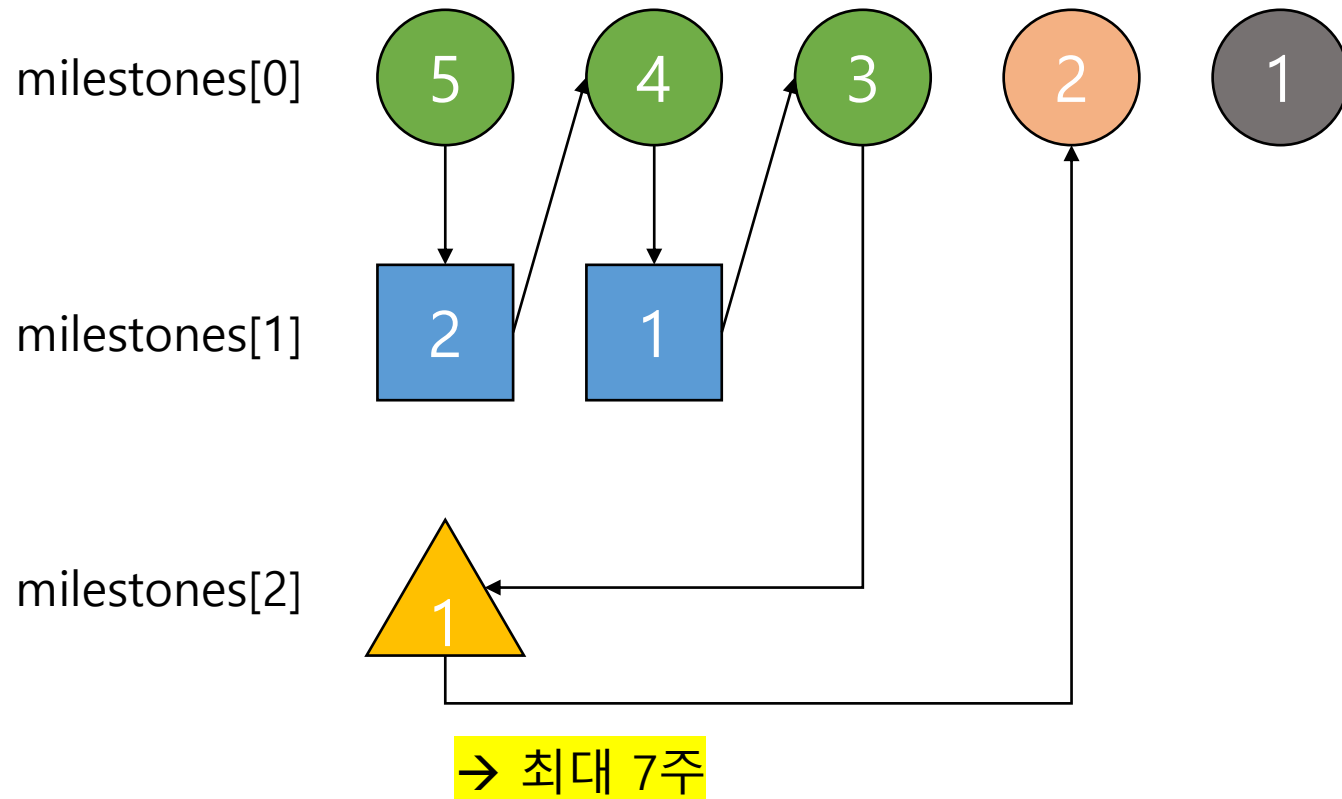
Step1 - 접근

milestones = [4, 2, 1]



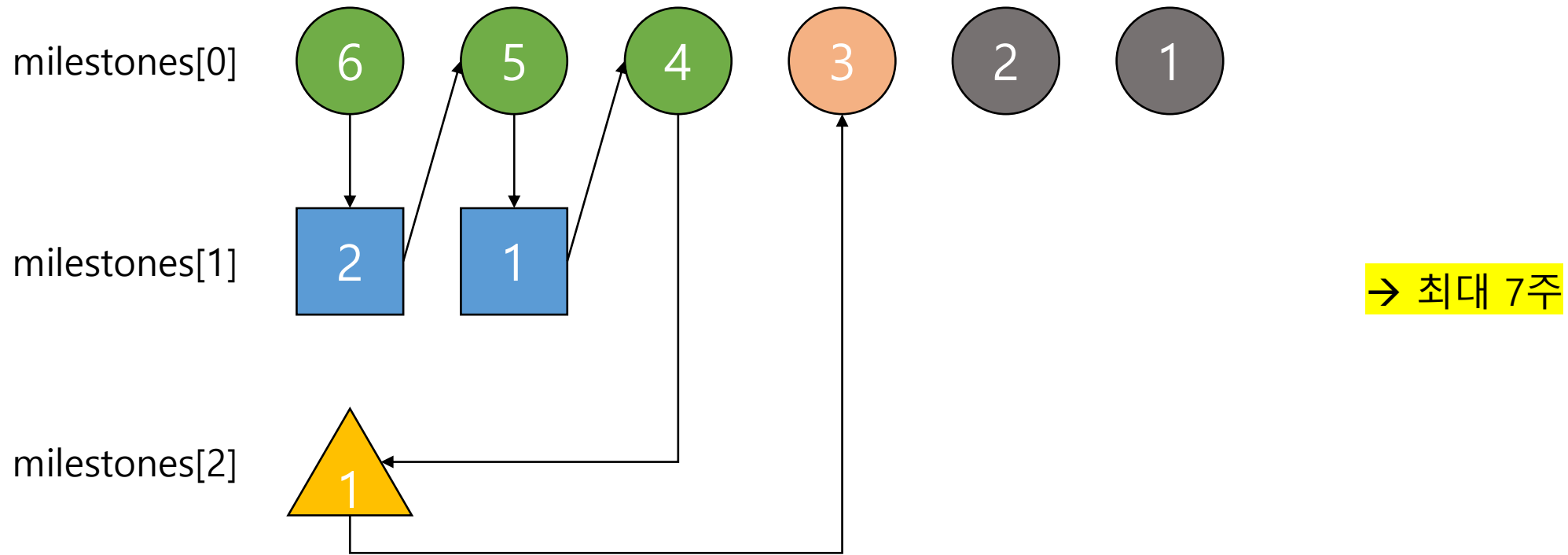
Step1 - 접근

milestones = [5, 2, 1]



Step1 – 접근

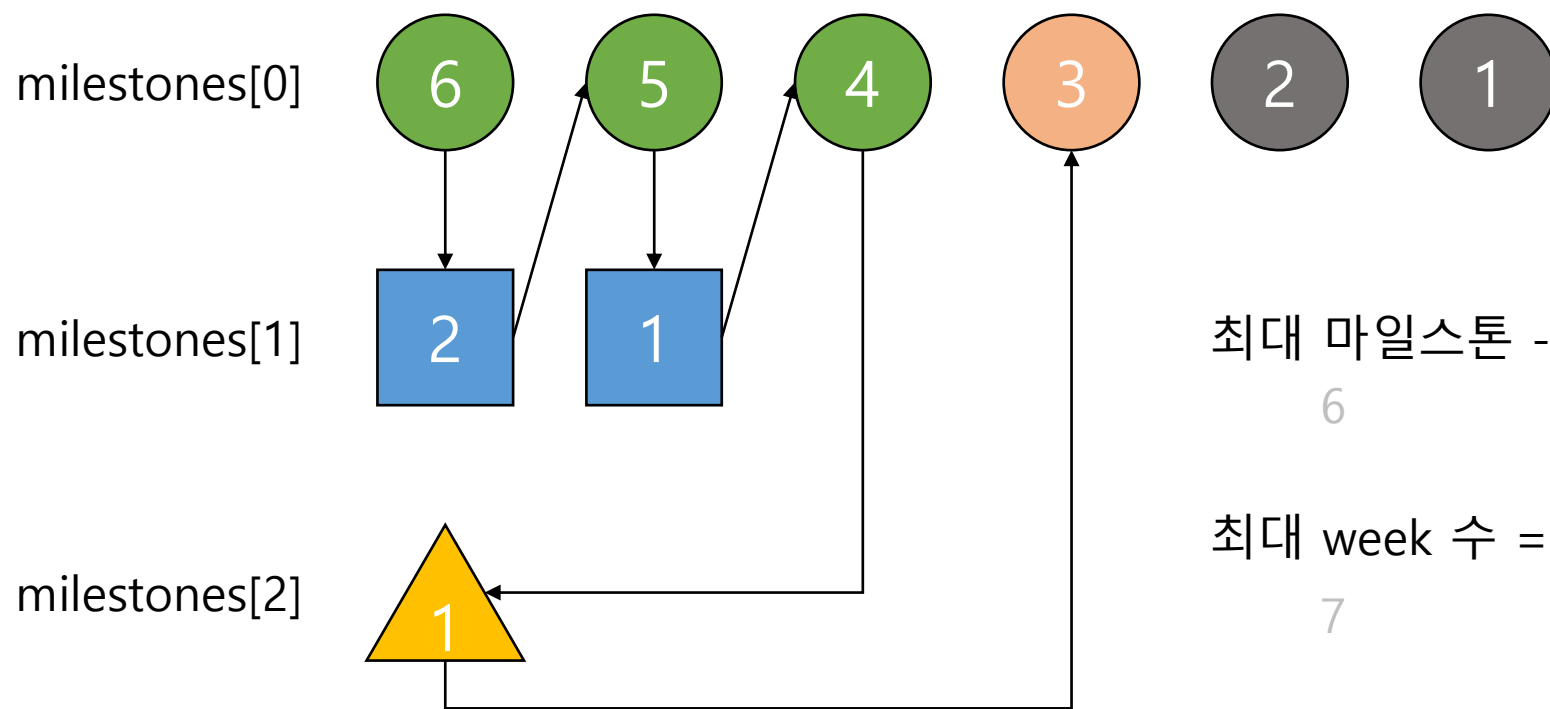
milestones = [6, 2, 1]



가장 큰 마일스톤의 프로젝트가 아닌 나머지 프로젝트들이 전체 week 수에 영향을 미침

Step2 – 관계 파악

milestones = [6, 2, 1]



최대 마일스톤 - 나머지 마일스톤의 합 > 0

6 (2 + 1)

최대 week 수 == 나머지 마일스톤의 합 * 2 + 1

7 (2 + 1)

왔다갔다 하는 값(?)

최대에서 한 번 더 일을 진행

Step2 – 관계 파악

① 최대 마일스톤 - 나머지 마일스톤의 합 > 0 인 경우, ② 최대 마일스톤 - 나머지 마일스톤의 합 == 0 인 경우,

최대 마일스톤 > 나머지 마일스톤의 합

milestones = [3, 2, 1]

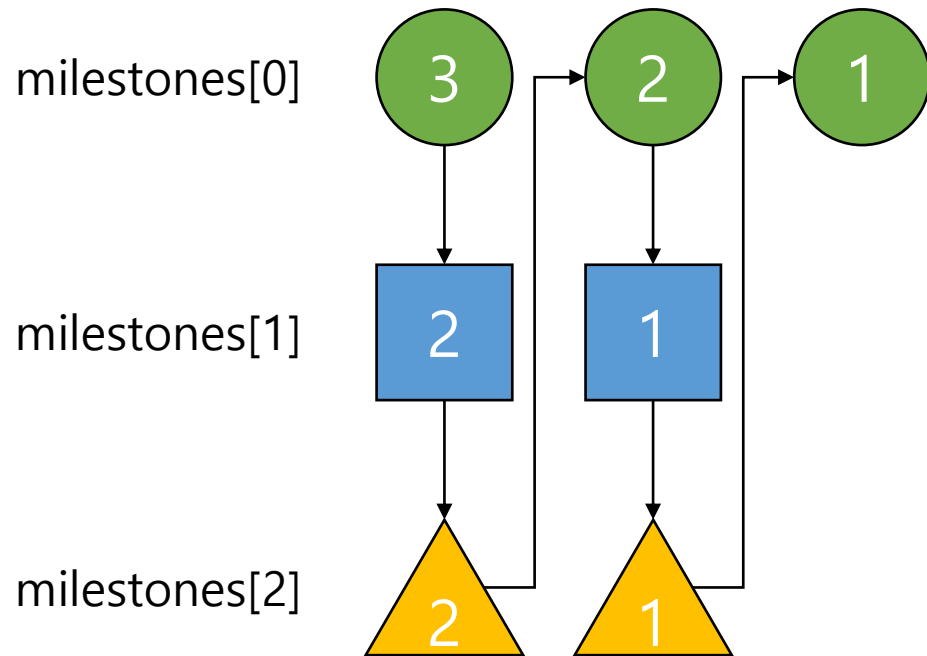
최대 마일스톤 > 전체 마일스톤의 합 - 최대 마일스톤 → 최대 week 수 = 전체 마일스톤의 합

→ 최대 week 수 = 나머지 마일스톤의 합 * 2 + 1

Step2 – 관계 파악

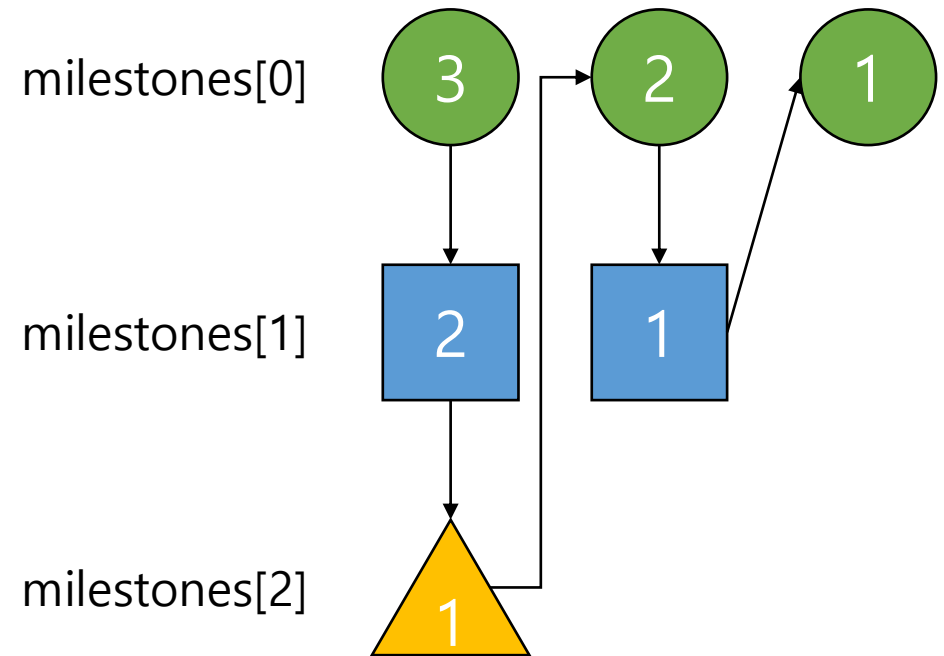
- ③ 최대 마일스톤 - 나머지 마일스톤의 합 < 0 인 경우, ② 최대 마일스톤 - 나머지 마일스톤의 합 == 0 인 경우,

milestones = [3, 2, 2]



→ 최대 7주

milestones = [3, 2, 1]



②, ③ 의 경우
최대 week 수 = 전체 마일스톤의 합

Step2.5 – 보너스

★ 최대 마일스톤을 가진 프로젝트가 2개 이상인 경우,

milestones = [1, 1, 1]

→ 최대 week 수 = 3

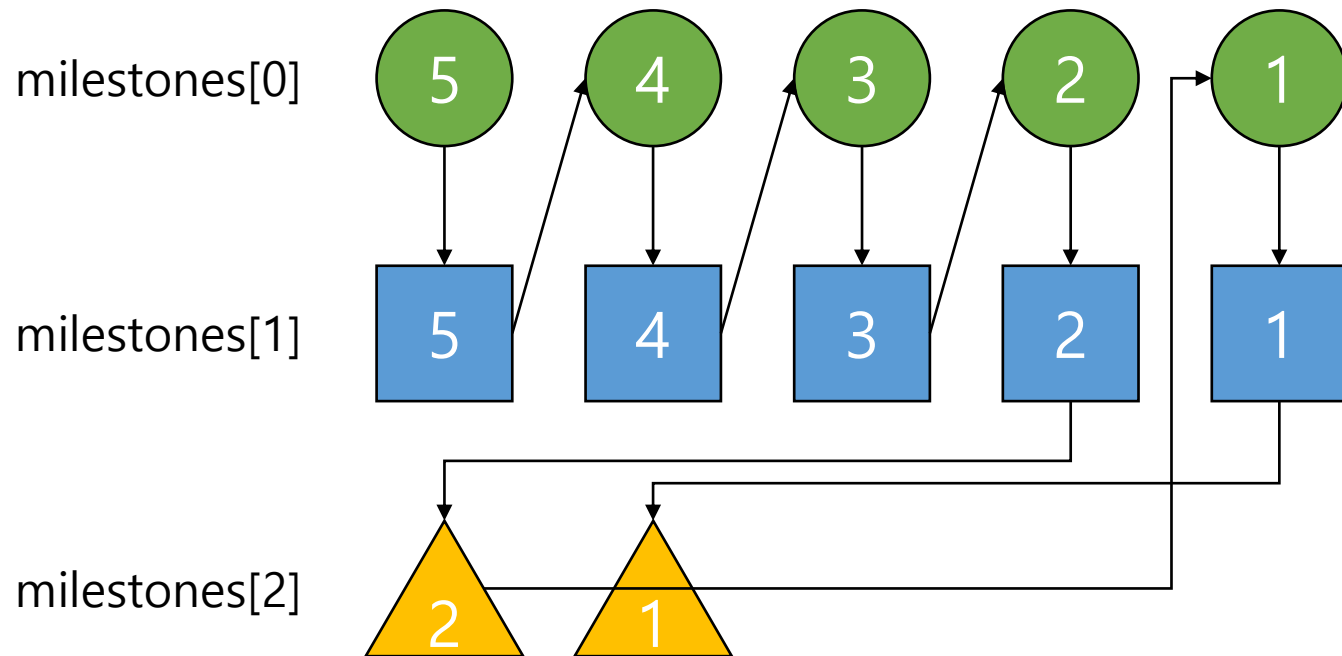
milestones = [2, 2, 2] → milestones = [1, 1, 1]

→ 최대 week 수 = 6

Step2.5 – 보너스

★ 최대 마일스톤을 가진 프로젝트가 2개 이상인 경우,

milestones = [5, 5, 2] → milestones = [4, 4, 2] → milestones = [3, 3, 2] → milestones = [2, 2, 2]



→ 최대 week 수 = 전체 마일스톤의 합

→ 최대 12주

Step3 – 구현

```
class Solution:
    def numberOfWeeks(self, milestones: List[int]) -> int:
        total_milestones, max_milestone = sum(milestones), max(milestones)

        if total_milestones - max_milestone < max_milestone:
            return 2 * (total_milestones - max_milestone) + 1

        return total_milestones
```

Thank you~!@#\$\$%^&*()_+