

Greedy 알고리즘

무지의 먹방 라이브 -

<https://school.programmers.co.kr/learn/courses/30/lessons/42891>

문제 설명

회전판에 먹어야 할 N개의 음식이 있다. 음식에는 1부터 N까지 번호가 붙어있다.

1. 무지는 먹방을 진행하면서 음식을 **1번부터 순서대로** 먹는다. (1번 → 2번 → 3번 → ... → N번 → 1번)
2. 각 음식에는 먹는데 걸리는 시간이 있으며, **1초에 1만큼의 음식을 먹는다.**
3. 무지는 음식 하나를 1초 동안 섭취한 후 남은 음식은 그대로 두고, 다음 음식을 섭취한다.
4. 회전판이 다음 음식을 무지 앞으로 가져오는데 걸리는 시간은 없다고 가정한다.

무지가 먹방을 시작한 지 K 초 후에 네트워크 장애로 인해 방송이 잠시 중단되었다.

무지는 네트워크 정상화 후 다시 방송을 이어갈 때, 몇 번 음식부터 섭취해야 하는지를 알고자 한다.

food_times	k	result
[3, 1, 2]	5	1

- **food_times** = 각 음식을 모두 먹는데 필요한 시간이 음식의 번호 순서대로 들어있는 배열
- **K** = 방송이 중단된 시간
- 섭취해야할 음식이 없으면 -1 반환

무지의 먹방 라이브 - 해결방법

📌 시간이 적게 걸리는 음식부터 확인하자 !!

- 모든 음식을 **시간 기준으로 정렬**한 후, 시간이 적게 걸리는 음식부터 제거해 나가는 방식 이용



1) 8초 소요



2) 6초 소요

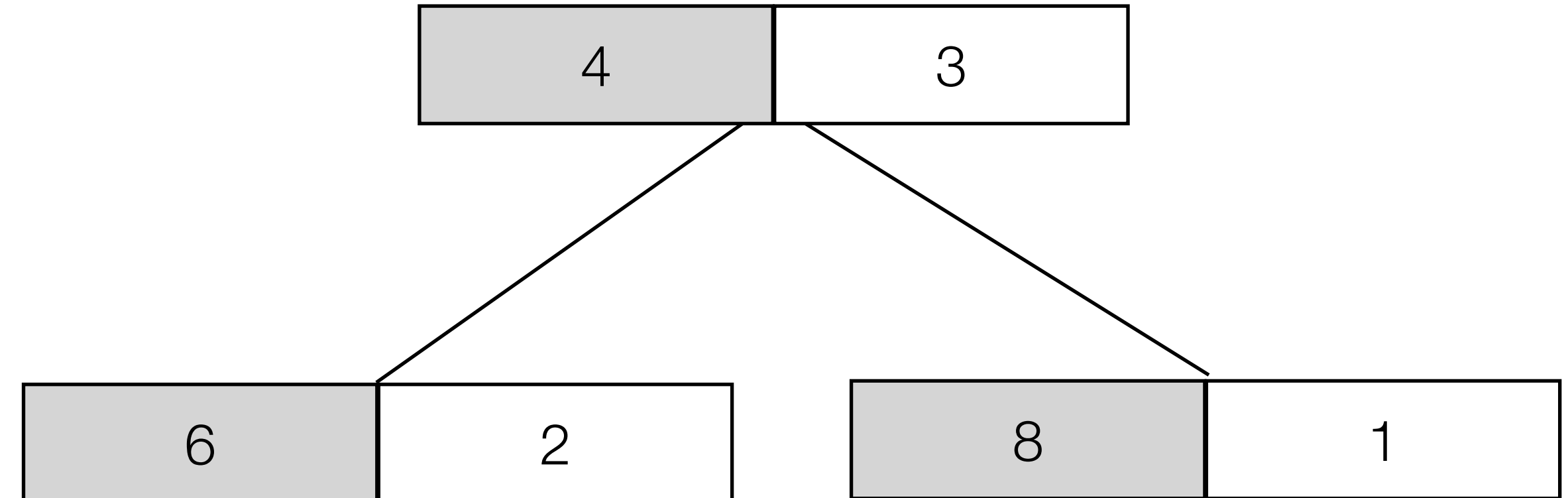


3) 4초 소요

전체 남은 시간(K) : 15초

남은 음식 : 3개

📌 Step1. 모든 음식을 priority Queue (min heap)에 삽입



(음식시간, 음식번호) 순

무지의 먹방 라이브 - 해결방법

📌 시간이 적게 걸리는 음식부터 확인하자 !!

- 모든 음식을 **시간 기준으로 정렬**한 후, 시간이 적게 걸리는 음식부터 제거해 나가는 방식 이용



1) 8초 소요



2) 6초 소요



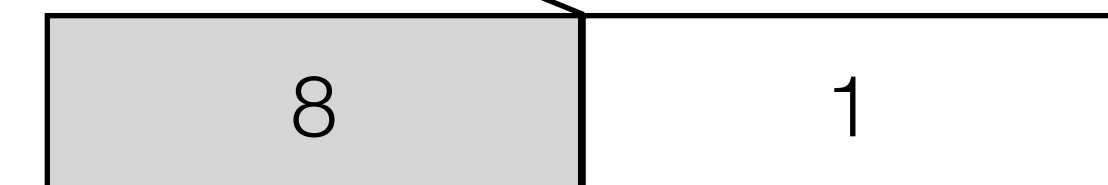
3) 4초 소요

📌 Step2. 첫번째 단계에서 시간이 가장 적게 걸리는 3번 음식을 뺀다.

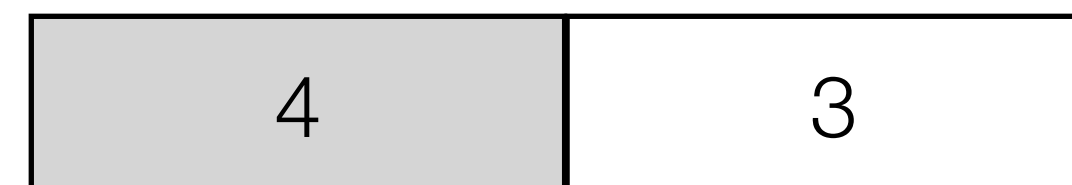
음식이 3개 남아있었으므로 $15 - 3 * 4$ 의 연산이 수행된다. ⇒ **전체 남은 시간 (K) : 3초**

전체 남은 시간(K) : 3초

남은 음식 : 2개



먹은 음식 :



무지의 먹방 라이브 - 해결방법

📌 시간이 적게 걸리는 음식부터 확인하자 !!

- 모든 음식을 **시간 기준으로 정렬**한 후, 시간이 적게 걸리는 음식부터 제거해 나가는 방식 이용



1) 8초 소요 2) 6초 소요 3) 4초 소요

📌 Step3. 전체 남은 시간 : 3초 / 이번 단계에서는 2번 음식을 빼야한다.

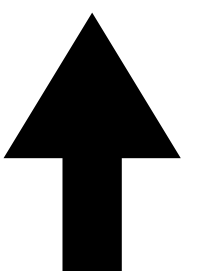
$2(\text{남은 음식 개수}) * 2(\text{음식 다 먹는 시간}) = 4\text{초} > \text{남은 시간 : 3초} \Rightarrow \text{빼지 않는다}$

Solution : 이제 '다음으로 먹어야 할 음식'을 찾아야 한다. (네번째 음식의 번호 출력)

전체 남은 시간(K) : 3초

남은 음식 : 2개

6	2	8	1	6	2	8	1
---	---	---	---	---	---	---	---



무지의 먹방 라이브 - 해결방법

📌 시간이 적게 걸리는 음식부터 확인하자 !!

- 모든 음식을 **시간 기준으로 정렬**한 후, 시간이 적게 걸리는 음식부터 제거해 나가는 방식 이용



1) 8초 소요 2) 6초 소요 3) 4초 소요

📌 Step3. 전체 남은 시간 : 3초 / 이번 단계에서는 2번 음식을 빼야한다.

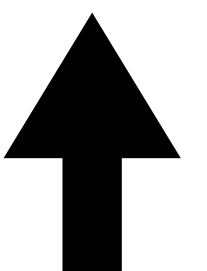
$2(\text{남은 음식 개수}) * 2(\text{음식 다 먹는 시간}) = 4\text{초} > \text{남은 시간 : 3초} \Rightarrow \text{빼지 않는다}$

Solution : 이제 '다음으로 먹어야 할 음식'을 찾아야 한다. (네번째 음식의 번호 출력)

전체 남은 시간(K) : 3초

남은 음식 : 2개

6	2	8	1	6	2	8	1
---	---	---	---	---	---	---	---



무지의 먹방 라이브 - 해결방법

```
#무지의 먹방 라이브
# https://school.programmers.co.kr/learn/courses/30/lessons/42891?language=python3

def solution(food_times, k):
    n = len(food_times)
    idx = 0

    for _ in range(k):
        # 아직 남은 음식이 없으면 -1
        if sum(food_times) == 0:
            return -1

        # 현재 음식 다 먹었으면 다음으로 넘기기
        while food_times[idx] == 0:
            idx = (idx + 1) % n

        # 1초 먹기
        food_times[idx] -= 1

        # 다음 음식으로
        idx = (idx + 1) % n

    # k초 후 아직 남은 음식이 없으면 -1
    if sum(food_times) == 0:
        return -1

    # k초 후 현재 idx에서 먹을 음식 찾기
    while food_times[idx] == 0:
        idx = (idx + 1) % n

    return idx + 1 # 음식 번호는 1부터 시작
```