

06

# 탐욕 알고리즘 이진 탐색 알고리즘

01 탐욕 알고리즘

02 이진 탐색 알고리즘

# 탐욕 알고리즘

---

- 탐욕(greedy) 알고리즘은 현재 상태에서 최선의 선택지가 전체에서 최선의 선택지가 되는 알고리즘
- Ex) 2820원을 동전으로 교환할 때, 동전의 개수가 최소가 되도록 동전을 선택하세요.
  - 동전의 종류 : 500원, 100원, 50원, 10원
  - 방법1. 금액이 큰 동전부터 최대한 많이 선택
    - 500원 5개, 100원 3개, 10원 2개, 총 10개의 동전 필요
  - 방법2. 금액이 작은 동전부터 최대한 많이 선택
    - 10원 282개, 총 282개의 동전 필요
  - 방법3. 금액이 큰 동전부터 1개씩 선택
    - 500원 4개, 100원 5개, 50원 5개, 10원 7개, 총 21개의 동전 필요

# 탐욕 알고리즘

---

- 방법1. 금액이 큰 동전부터 최대한 많이 선택 - 증명
  - 총 금액 M원, 500원 x개, 100원 y개, 50원 z개, 10원 w개
  - 가설에 의해  $x = M // 500$ ,  $y = (M - 500 * x) // 100$ ,  $z = \dots$
  - 가설이 거짓이라 가정하면,
    - Case1.  $\exists x'$  such that  $x' \neq x$
    - 이하 생략
- 설정한 방법을 증명하기 보다 **반례**를 빠르게 찾는 것이 중요

# 탐욕 알고리즘

---

- 방법2. 금액이 작은 동전부터 최대한 많이 선택
  - 10원 282개, 총 282개의 동전 필요
  - 50원 1개, 10원 277개, 총 278개로 282개보다 적은 수의 동전 선택 가능 → 잘못된 방법
- 방법3. 금액이 큰 동전부터 1개씩 선택
  - 500원 4개, 100원 5개, 50원 5개, 10원 7개, 총 21개의 동전 필요
  - 500원 5개, 50원 5개, 10원 7개, 총 17개로 21개보다 적은 수의 동전 선택 가능 → 잘못된 방법

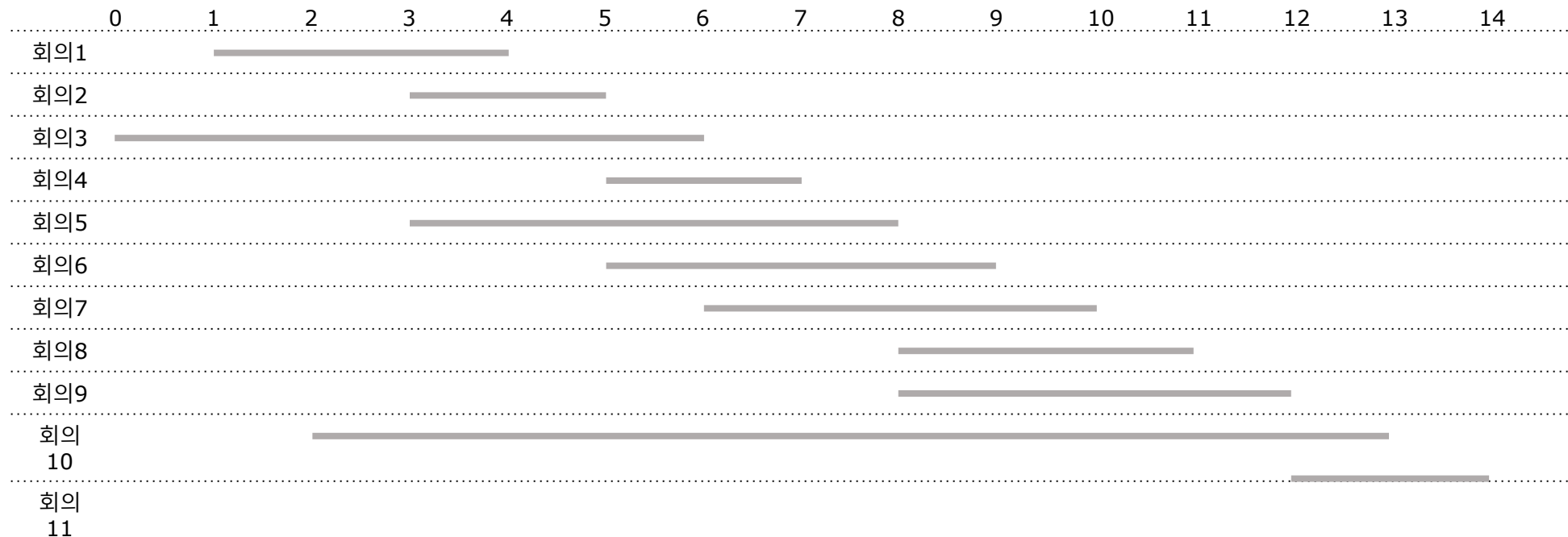
# 연습문제

---

- 백준 온라인 저지 - 11047 동전 0
- 백준 온라인 저지 - 2839 설탕 배달

# 탐욕 알고리즘

- Ex) 한 개의 회의실에 다음 11개의 회의에 대해 시간이 겹치지 않게 최대한 많은 회의를 배정하세요. 각 회의는 시작시간, 종료시간이 주어집니다.
  - (시작, 종료) : (1, 4), (3, 5), (0, 6), (5, 7), (3, 8), (5, 9), (6, 10), (8, 11), (8, 12), (2, 13), (12, 14)



# 탐욕 알고리즘

---

- 방법1. 회의 시간이 가장 짧은 회의부터 선택



- 방법2. 가장 먼저 시작하는 회의부터 선택



# 탐욕 알고리즘

---

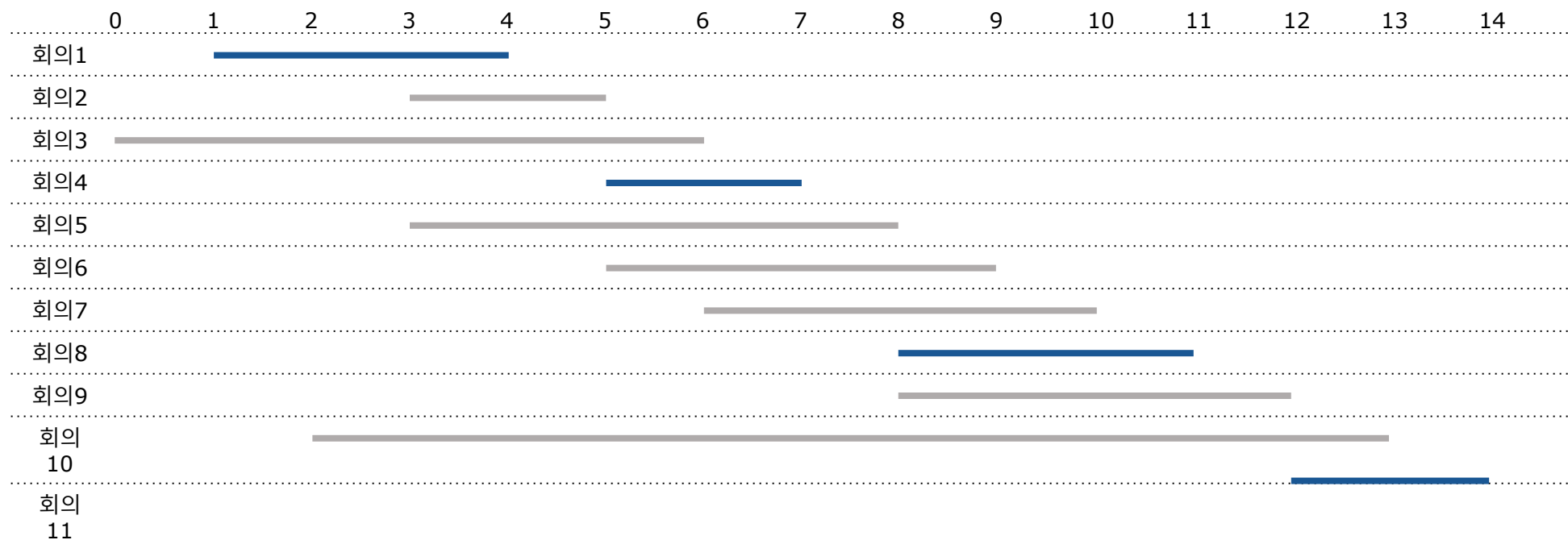
- 방법3. 가장 먼저 끝나는 회의부터 선택





# 탐욕 알고리즘

- 방법3. 가장 먼저 끝나는 회의부터 선택
  - (시작, 종료) : (1, 4), (3, 5), (0, 6), (5, 7), (3, 8), (5, 9), (6, 10), (8, 11), (8, 12), (2, 13), (12, 14)



# 탐욕 알고리즘

---

- pseudo-code

```
N // 회의의 수
meetings // 입력 받은 회의 배열
result // 선택한 회의

meetings.sort() // 종료시간, 시작시간 기준 정렬
result.add(meetings[0])

for i = 1 to N-1 :
    if (result의 마지막 회의의 종료시간 <= meetings[i]의 시작시간)
        then result.add(meetings[i])
```

- 시간복잡도  $O(n \log n)$

# 연습문제

---

- 백준 온라인 저지 - 1931 회의실 배정

# 이진 탐색

## ■ 이진 탐색(binary search)

- 데이터가 정렬되어 있는 상태에서 원하는 값을 찾아내는 알고리즘
- 배열의 중앙값과 원하는 값을 비교해가며 탐색
- 정렬된 배열에서 탐색의 시간복잡도  $O(\log n)$

## ■ pseudo-code

```
arr // 배열  
target // 검색할 값  
start = 0, end = N-1  
arr.sort()  
  
while start <= end  
    center = (start+end)/2  
    if arr[center] < target then start=center+1  
    if arr[center] > target then end=center-1  
    if arr[center] == target then print(center)
```



이진 탐색 과정

# 연습문제

---

- 1부터 1000까지의 정수 배열을 생성하고 800을 이진 탐색하세요.
  - 800이 존재하는 경우 해당 인덱스를 출력, 존재하지 않으면 -1 출력
  - 탐색 횟수 출력

# 연습문제

---

- 백준 온라인 저지 - 1920 수 찾기

# 연습문제

---

- 백준 온라인 저지 - 2630 색종이 만들기