

JavaScript

탐욕법 알고리즘

4) 그리디 문제 풀이 ③

그리디 문제 풀이 | 코딩 테스트에서 자주 등장하는 탐욕법 알고리즘 이해하기

강사 나동빈

JavaScript

탐욕법 알고리즘

4) 그리디 문제 풀이 ③

JavaScript 탐욕법
그리디 문제 풀이

혼자 힘으로 풀어보기

JavaScript
탐욕법
그리디
문제 풀이

문제 제목: 주유소

문제 난이도: ★★☆☆☆

문제 유형: 그리디

추천 풀이 시간: 30분

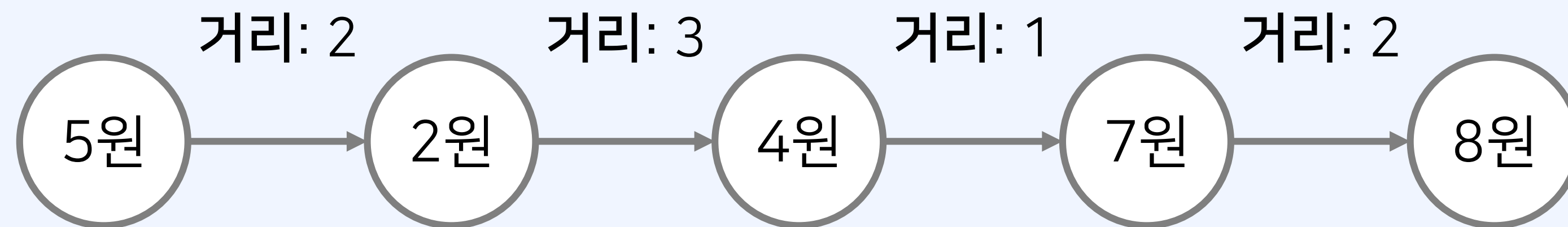
JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

JavaScript
탐욕법
그리디
문제 풀이

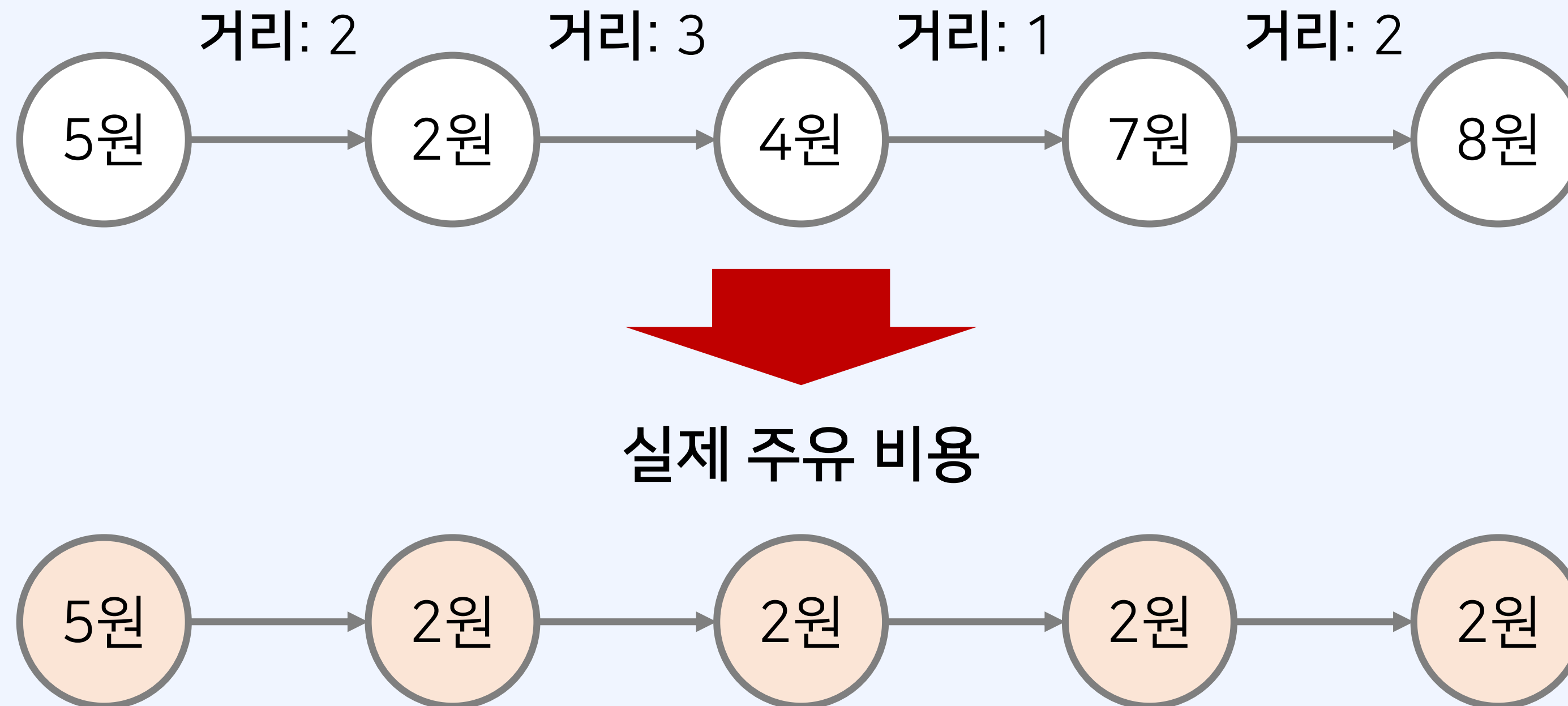
- 예시를 확인해 보며 문제 해결을 위한 아이디어를 생각해 보자.
- 거리: [2, 3, 1, 2]
- 주유 비용: [5, 2, 4, 7, 8]

- 예시를 확인해 보며 문제 해결을 위한 아이디어를 생각해 보자.
- 거리: [2, 3, 1, 2]
- 주유 비용: [5, 2, 4, 7, 8]



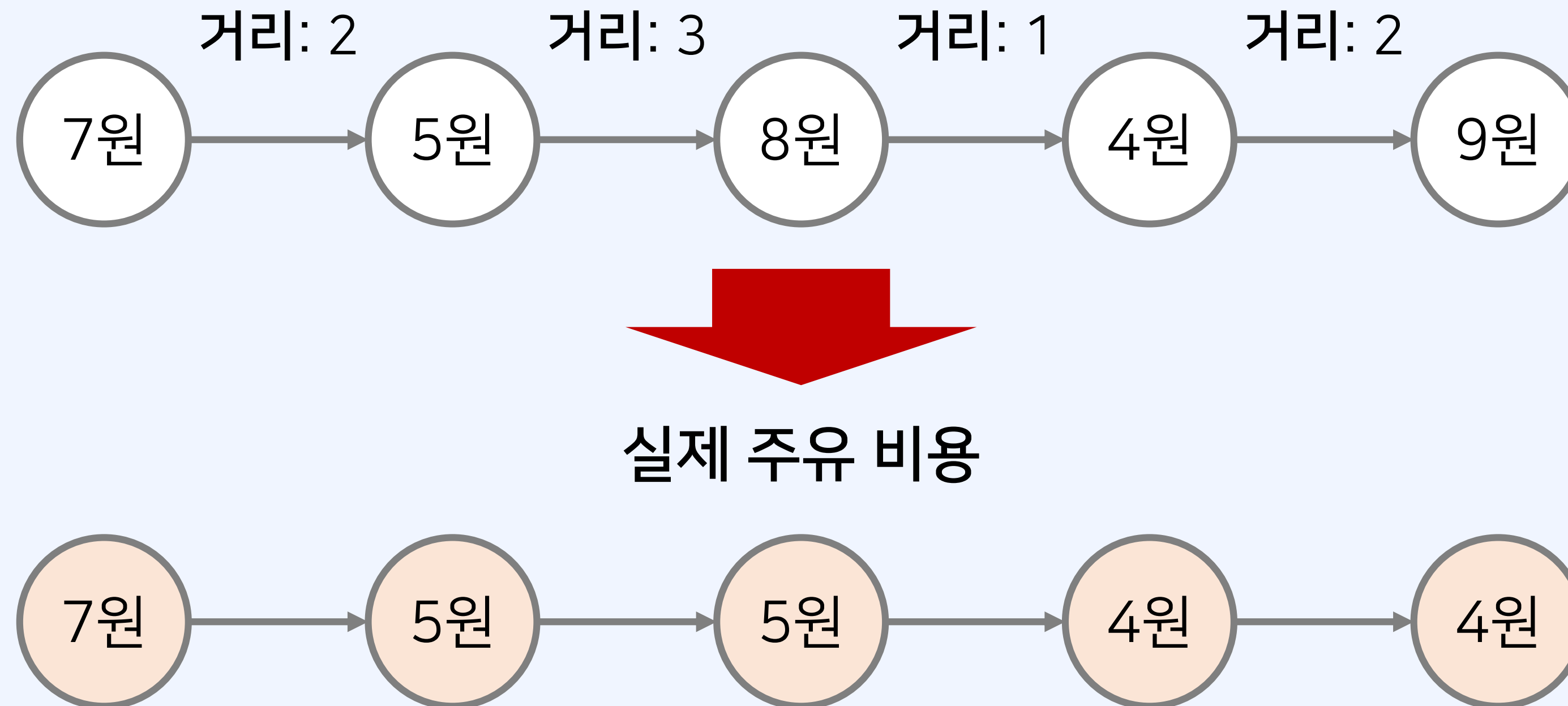
[핵심 아이디어] 주유 비용을 비오름차순으로 변경한다.

- 자기보다 뒤에 있는 비싼 주유소에 대해서 미리 결제하는 것으로 이해할 수 있다.
- 예시 1)



[핵심 아이디어] 주유 비용을 비오름차순으로 변경한다.

- 자기보다 뒤에 있는 비싼 주유소에 대해서 미리 결제하는 것으로 이해할 수 있다.
- 예시 2)



JavaScript 탐욕법 그리디 문제 풀이

정답 코드 예시

JavaScript 탐욕법 그리디 문제 풀이

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let n = Number(input[0]);
let dist = input[1].split(' ').map(Number);
let cost = input[2].split(' ').map(Number);

// 주유 비용(cost) 배열의 값을 비오름차순이 되도록 변환
// [5, 2, 4, 1] -> [5, 2, 2, 1]
let minCost = cost[0];
for (let i = 0; i < n; i++) {
    minCost = Math.min(minCost, cost[i]);
    cost[i] = minCost;
}

// 도로당 이동 비용의 합 계산
let answer = BigInt(0);
for (let i = 0; i < n - 1; i++) {
    // JavaScript에서 큰 정수를 처리할 때는 BigInt 사용
    answer += BigInt(dist[i]) * BigInt(cost[i]);
}
console.log(String(answer)); // 뒤에 붙는 'n' 제거
```


JavaScript 탐욕법
그리디 문제 풀이

혼자 힘으로 풀어보기

JavaScript
탐욕법
그리디
문제 풀이

문제 제목: 회의실 배정

문제 난이도: ★★☆☆☆

문제 유형: 그리디

추천 풀이 시간: 40분

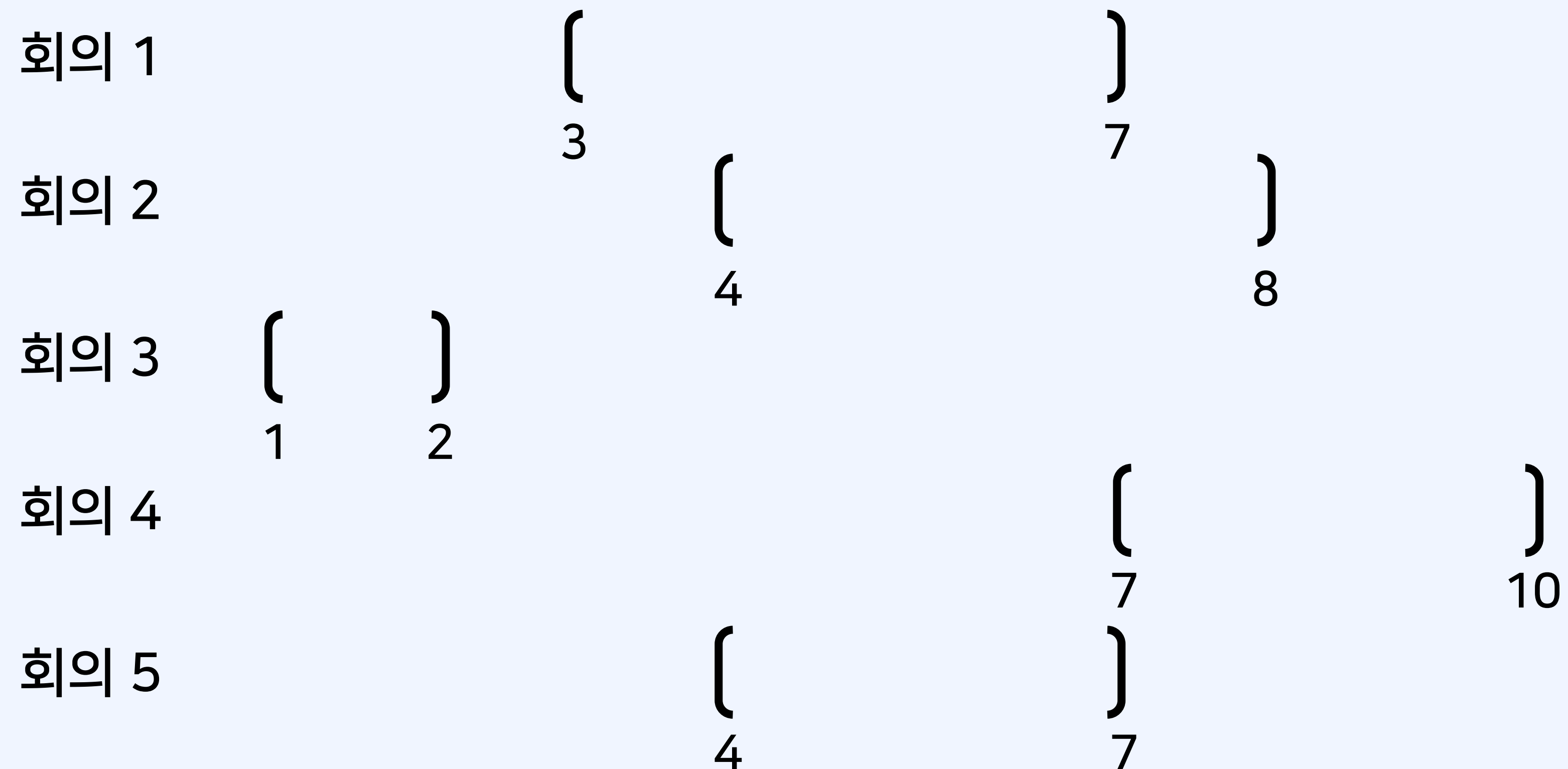
- 문제의 요구 사항은 다음과 같다.
- 문제 요약: 각 회의가 겹치지 않게 하면서 회의실을 사용할 수 있는 회의의 최대 개수 계산

[문제 해결 아이디어]

- 가장 먼저 모든 회의에 대하여 오름차순 정렬한다.
- 정렬할 때는 1) 종료 시점 2) 시작 시점을 우선순위로 한다.
- 첫 번째 회의부터 시작해 겹치지 않게 최대한 많은 회의를 선택한다.

- 예시를 확인해 보자.

[핵심] 종료 시점이 이른 회의부터 확인하며, 겹치지 않게 배정한다.



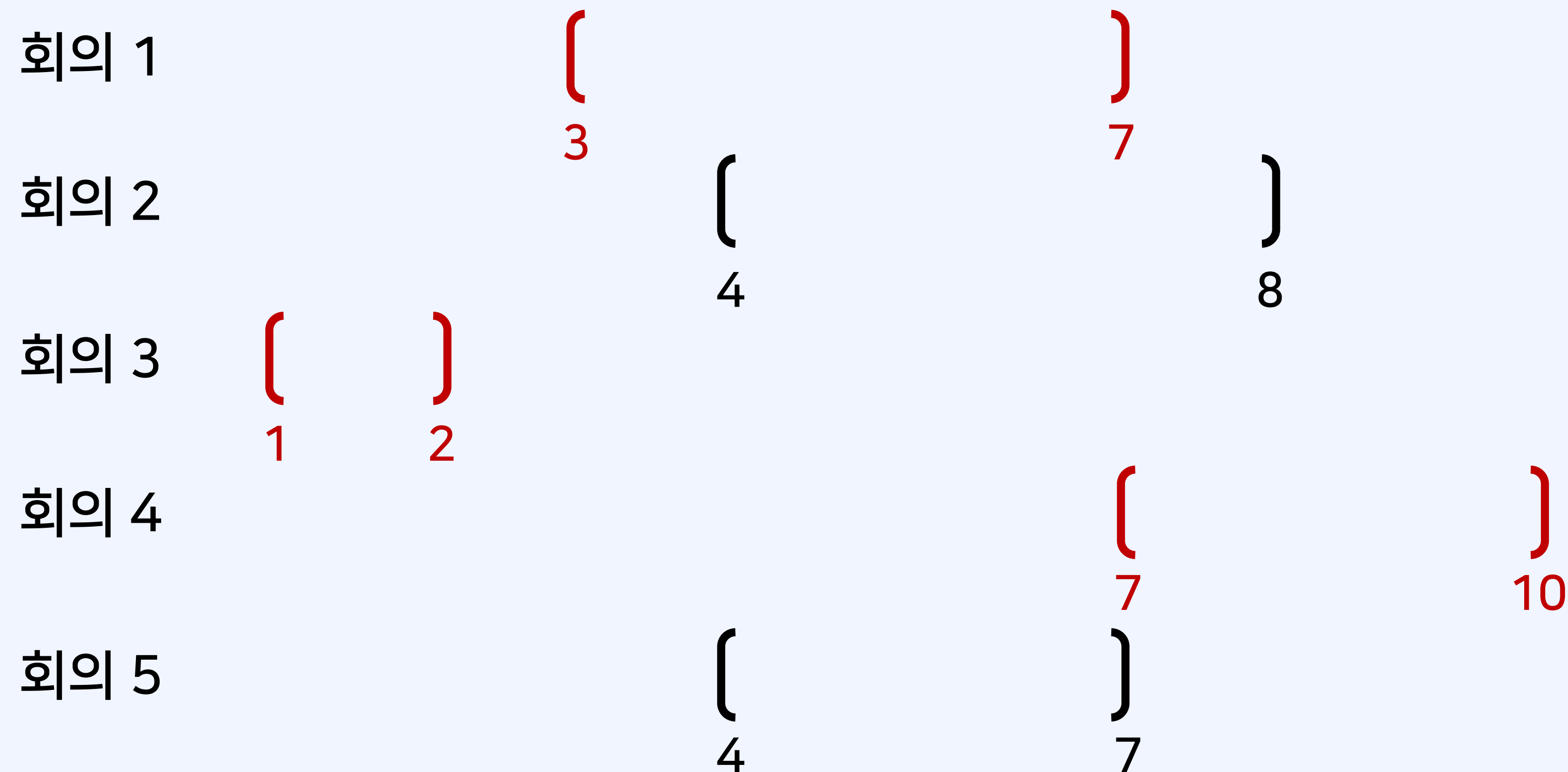
JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

JavaScript
탐욕법
그리디
문제 풀이

[핵심] 종료 시점이 이른 회의부터 확인하며, 겹치지 않게 배정한다.

- 결과적으로 세 개의 회의를 배치할 수 있다. (정답: 3개)



JavaScript 탐욕법 그리디 문제 풀이

정답 코드 예시

JavaScript 탐욕법 그리디 문제 풀이

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let n = Number(input[0]); // 회의의 개수
let arr = []; // 각 회의에 대한 (시작 시점, 종료 시점) 입력받기
for (let i = 1; i <= n; i++) arr.push(input[i].split(' ').map(Number));

arr.sort(function(a, b) { // (1) 종료 시점 (2) 시작 시점을 우선순위로 오름차순 정렬
  if (a[1] != b[1]) return a[1] - b[1];
  else return a[0] - b[0];
});

let cnt = 1, cur = 0; // 첫 번째 회의부터 확인
for (let i = 1; i < n; i++) {
  if (arr[cur][1] <= arr[i][0]) { // 현재 회의가 끝난 이후에 회의가 시작되는 경우 카운트
    cur = i;
    cnt += 1;
  }
}
console.log(cnt);
```

JavaScript 탐욕법
그리디 문제 풀이

혼자 힘으로 풀어보기

JavaScript
탐욕법
그리디
문제 풀이

문제 제목: 풍선 맞추기

문제 난이도: ★★☆☆☆

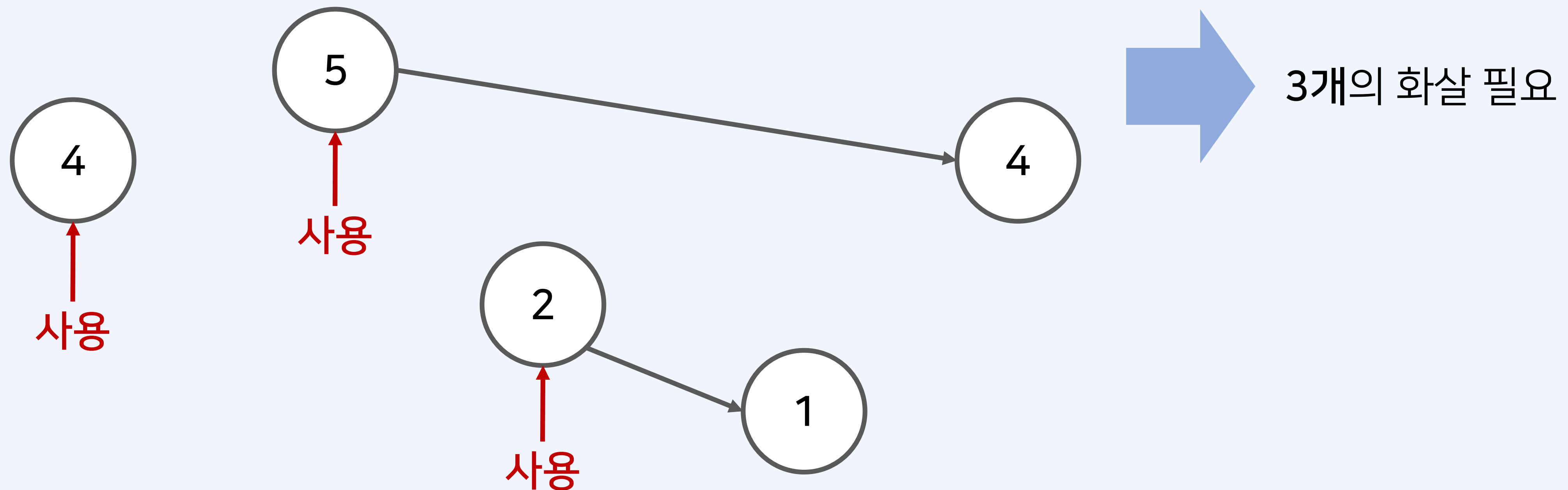
문제 유형: 그리디

추천 풀이 시간: 40분

- 왼쪽부터 하나씩 풍선을 확인한다.

[핵심 아이디어] 해당 높이에 존재하는 화살이 없다면 화살을 새롭게 사용한다.

- 5개의 풍선의 높이가 각각 [4, 5, 2, 1, 4]일 때는 다음과 같다.

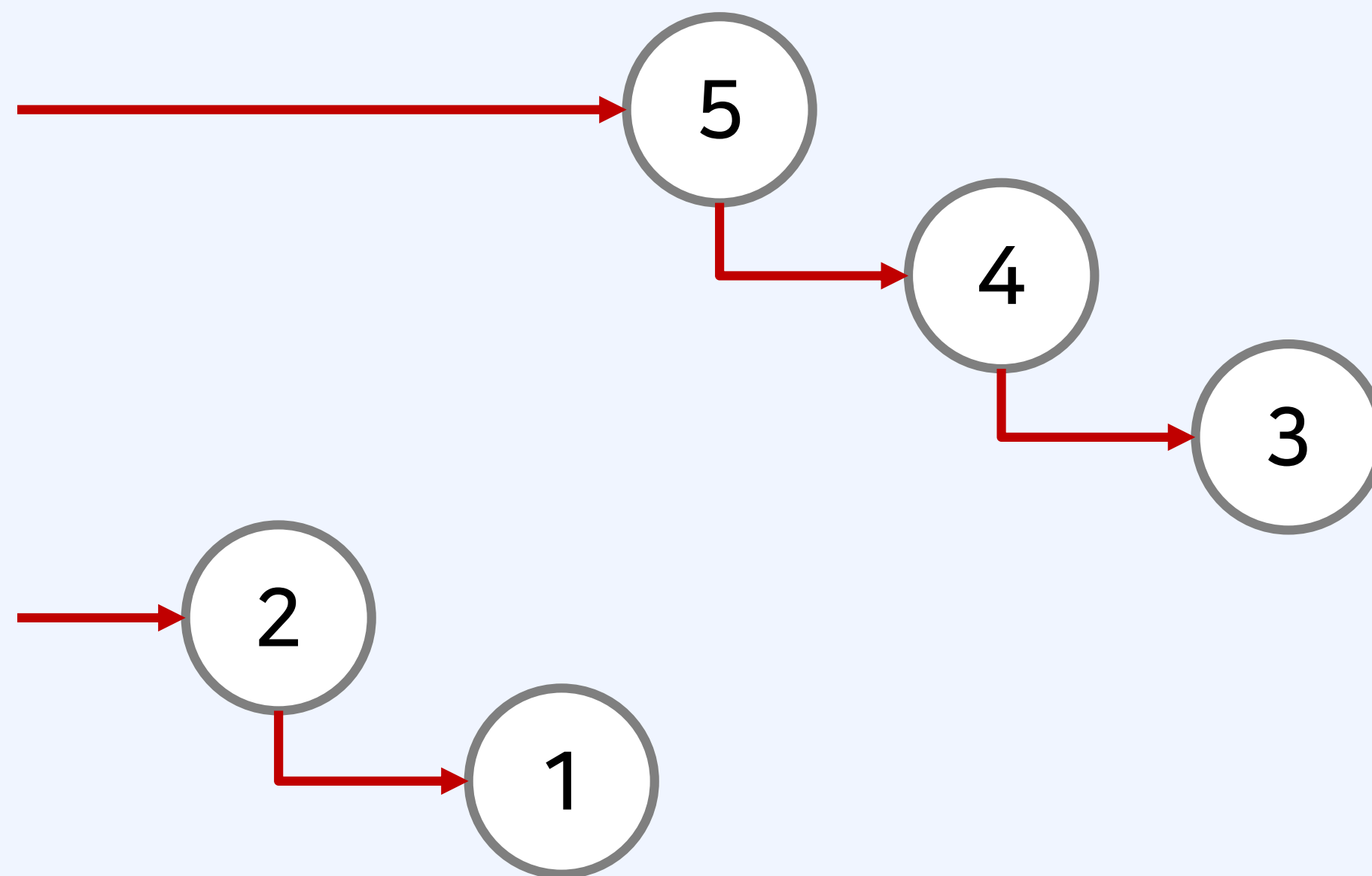


JavaScript 탐욕법 그리디 문제 풀이

문제 해결 아이디어

JavaScript
탐욕법
그리디
문제 풀이

- 예시를 통해 풀이 방법을 이해해 보자.
- 예시 1)



초기

높이	0	1	2	3	4	5
개수	0	0	0	0	0	0

② 처리
(+1)

높이	0	1	2	3	4	5
개수	0	1	0	0	0	0

① 처리

높이	0	1	2	3	4	5
개수	1	0	0	0	0	0

⑤ 처리
(+1)

높이	0	1	2	3	4	5
개수	0	0	0	0	1	0

④ 처리

높이	0	1	2	3	4	5
개수	1	0	0	1	0	0

③ 처리

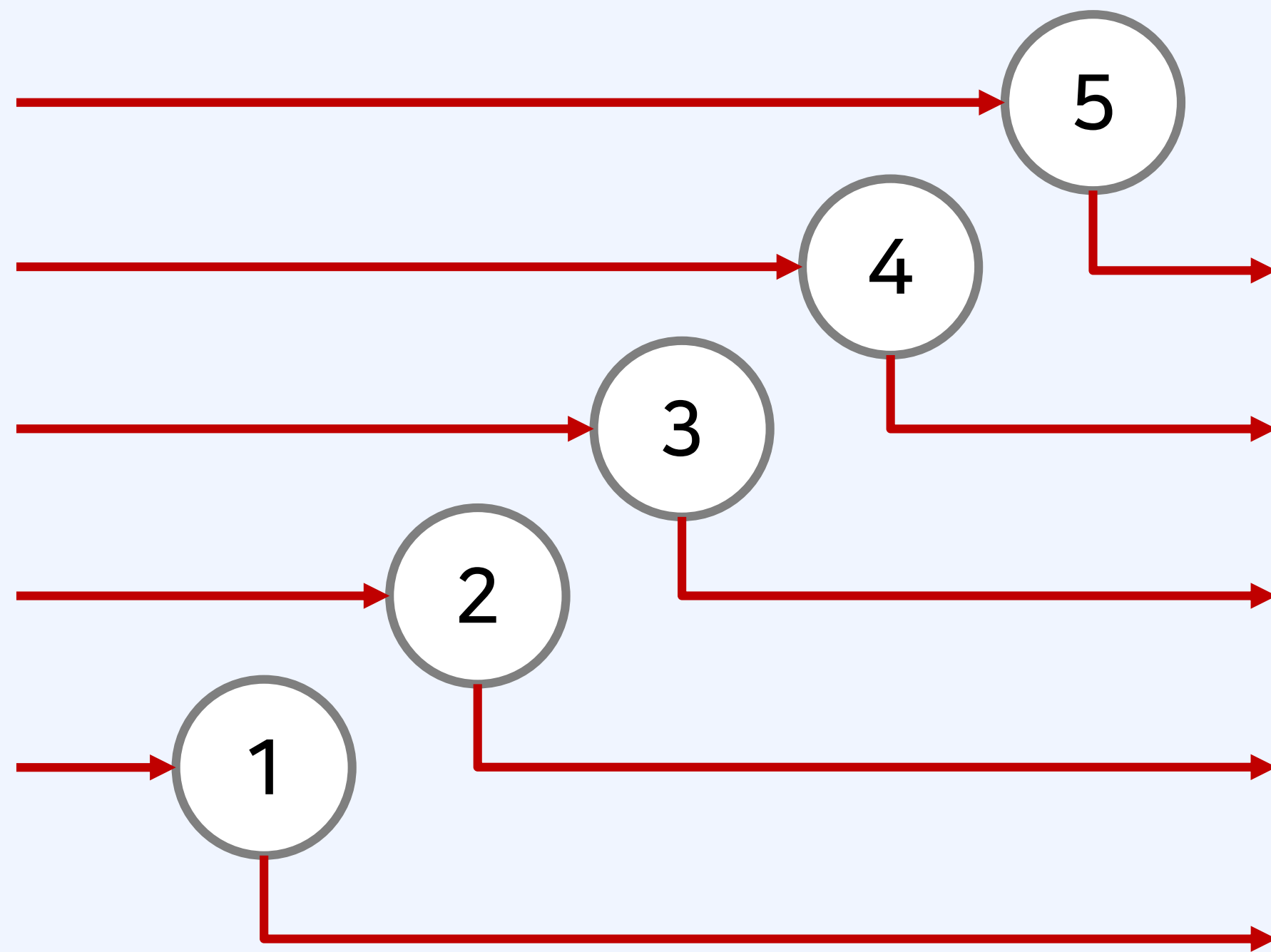
높이	0	1	2	3	4	5
개수	1	0	1	0	0	0

JavaScript 탐욕법 그리디 문제 풀이

문제 해결 아이디어

JavaScript 탐욕법 그리디 문제 풀이

- 예시를 통해 풀이 방법을 이해해 보자.
- 예시 2)**



초기

높이	0	1	2	3	4	5
개수	0	0	0	0	0	0

① 처리
(+1)

높이	0	1	2	3	4	5
개수	1	0	0	0	0	0

② 처리
(+1)

높이	0	1	2	3	4	5
개수	1	1	0	0	0	0

③ 처리
(+1)

높이	0	1	2	3	4	5
개수	1	1	1	0	0	0

④ 처리
(+1)

높이	0	1	2	3	4	5
개수	1	1	1	1	0	0

⑤ 처리
(+1)

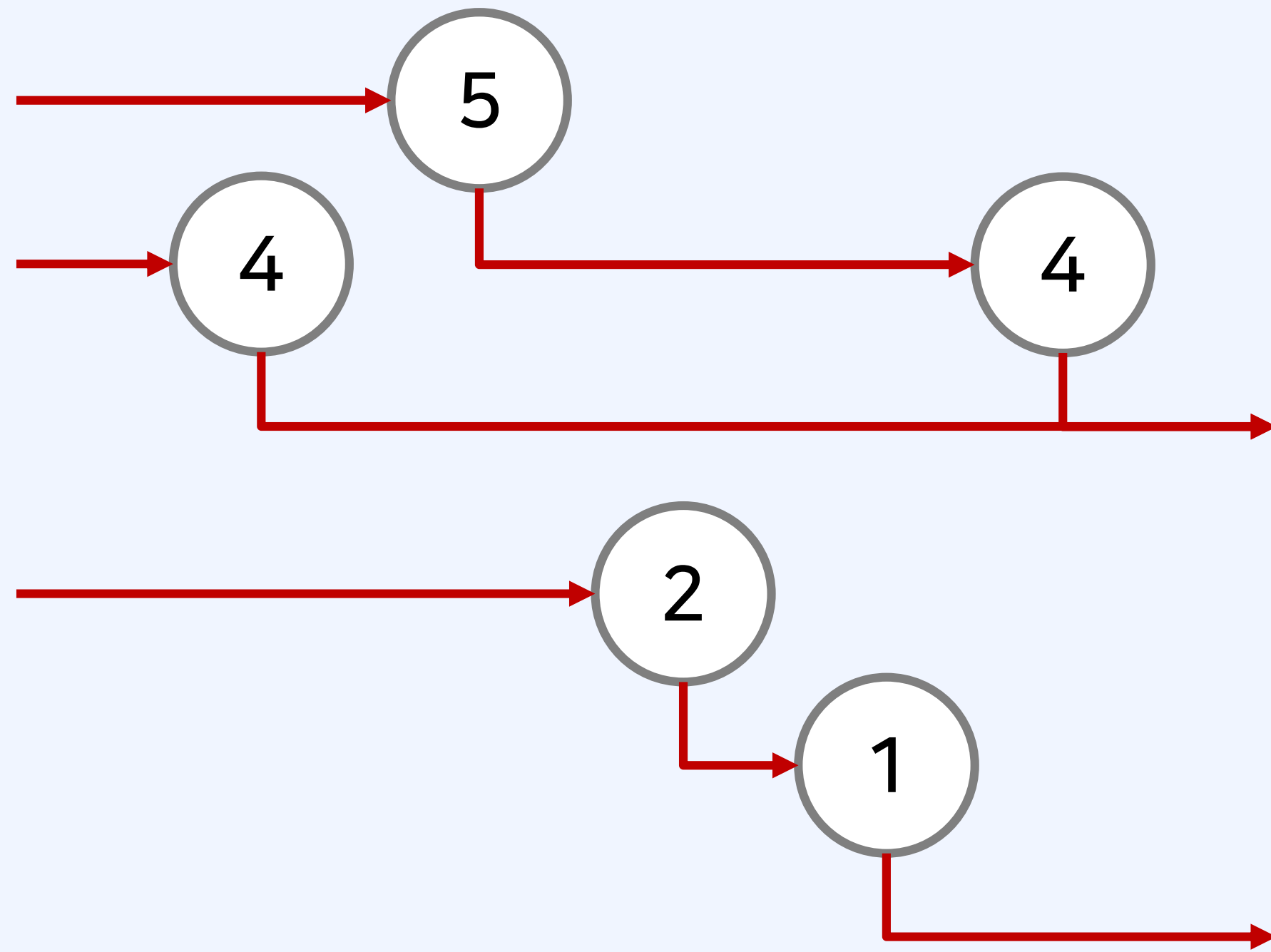
높이	0	1	2	3	4	5
개수	1	1	1	1	1	0

JavaScript 탐욕법 그리디 문제 풀이

문제 해결 아이디어

JavaScript 탐욕법 그리디 문제 풀이

- 예시를 통해 풀이 방법을 이해해 보자.
- 예시 3)



초기

높이	0	1	2	3	4	5
개수	0	0	0	0	0	0

④ 처리
(+1)

높이	0	1	2	3	4	5
개수	0	0	0	1	0	0

⑤ 처리
(+1)

높이	0	1	2	3	4	5
개수	0	0	0	1	1	0

② 처리
(+1)

높이	0	1	2	3	4	5
개수	0	1	0	1	1	0

① 처리

높이	0	1	2	3	4	5
개수	1	0	0	1	1	0

④ 처리

높이	0	1	2	3	4	5
개수	0	1	0	2	0	0

JavaScript 탐욕법 그리디 문제 풀이

정답 코드 예시

JavaScript
탐욕법
그리디
문제 풀이

```
const rl = require('readline').createInterface({
  input: process.stdin,
  output: process.stdout
});

let input = [];
rl.on('line', function(line) {
  input.push(line); // 콘솔 입력 창에서 줄바꿈(Enter)를 입력할 때마다 호출
}).on('close', function() {
  let data = input[1].split(' ').map(Number); // 모든 풍선의 위치 정보 입력받기
  let result = 0;
  let arrow = new Array(1000001).fill(0); // 각 높이에 화살이 몇 개 있는지
  for (let x of data) {
    if (arrow[x] > 0) { // 해당 높이에 화살이 있다면
      arrow[x] -= 1;
      arrow[x - 1] += 1;
    }
    else { // 해당 높이에 화살이 없다면
      arrow[x - 1] += 1;
      result += 1; // 화살 쏘기
    }
  }
  console.log(result);
  process.exit();
});
```

JavaScript 탐욕법
그리디 문제 풀이

혼자 힘으로 풀어보기

JavaScript
탐욕법
그리디
문제 풀이

문제 제목: 피보나치

문제 난이도: ★★☆☆☆

문제 유형: 그리디

추천 풀이 시간: 40분

JavaScript 탐욕법 그리디 문제 풀이

문제 해결 아이디어

JavaScript 탐욕법 그리디 문제 풀이

- **피보나치 수**: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...
- **명제**: 양의 정수는 하나 혹은 그 이상의 서로 다른 피보나치 수들의 합으로 나타낼 수 있다.
- 예를 들어, 정수 100은 다음과 같은 조합으로 표현할 수 있다.
- $100 = 3 + 8 + 89$
- $100 = 1 + 2 + 8 + 89$
- $100 = 3 + 8 + 34 + 55$

[문제 해결 방법] 가능한 가장 큰 피보나치 수부터 빼 나갈 때 최소 개수를 만족할 수 있다.

- **이유**: $X - A$ 는 A 보다 작은 서로 다른 피보나치 수들의 합으로 표현 가능하다.
- 이때 A 는 가능한 가장 큰 피보나치 수를 의미한다.

[문제 해결 방법] 가능한 가장 큰 피보나치 수부터 빼 나갈 때 최소 개수를 만족할 수 있다.

- 명제: 양의 정수는 하나 혹은 그 이상의 서로 다른 피보나치 수들의 합으로 나타낼 수 있다.

인덱스	0	1	2	3	4	5	6	7	8	9	10	11
값	0	1	1	2	3	5	8	13	21	34	55	89

1. 예시: $70 = 55 + 13 + 2$
2. 예시: $100 = 89 + 8 + 3$

JavaScript 탐욕법 그리디 문제 풀이

정답 코드 예시

JavaScript
탐욕법
그리디
문제 풀이

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

// 피보나치 수들 계산
pibo = [];
pibo.push(0);
pibo.push(1);
while (pibo[pibo.length - 1] < 1e9) pibo.push(pibo[pibo.length - 2] + pibo[pibo.length - 1]);

let testCases = Number(input[0]);
for (let tc = 1; tc <= testCases; tc++) {
  let n = Number(input[tc]);
  let result = [];
  let i = pibo.length - 1; // 가장 큰 피보나치 수의 인덱스
  while (n > 0) { // n이 0이 될 때까지
    if (n >= pibo[i]) { // 가능한 큰 피보나치 수부터 빼기
      n -= pibo[i];
      result.push(pibo[i]);
    }
    i--;
  }
  let answer = '';
  for (let i = result.length - 1; i >= 0; i--) answer += result[i] + ' '; // 오름차순 출력
  console.log(answer);
}
```