

# JavaScript 탐욕법 알고리즘

## 5) 그리디 문제 풀이 ④

그리디 문제 풀이 | 코딩 테스트에서 자주 등장하는 탐욕법 알고리즘 이해하기

강사 나동빈

# JavaScript

## 탐욕법 알고리즘

### 5) 그리디 문제 풀이 ④

JavaScript 탐욕법  
그리디 문제 풀이

## 혼자 힘으로 풀어보기

JavaScript  
탐욕법  
그리디  
문제 풀이

문제 제목: 박 터뜨리기

문제 난이도: ★★☆☆☆

문제 유형: 그리디

추천 풀이 시간: 40분

## JavaScript 탐욕법 그리디 문제 풀이

## 문제 해결 아이디어

## JavaScript 탐욕법 그리디 문제 풀이

- 각 바구니에 담긴 공의 개수가 모두 달라야 한다.
- 가장 많이 담긴 바구니와 가장 적게 담긴 바구니의 공의 개수 차이가 최소가 되려면?
- 바로 공의 개수가 **최대한 연속적**이게 만들면 된다.
  - 예를 들어  $N=15$ 이고  $K=5$ 일 때, 각 바구니에 공의 개수로 1, 2, 3, 4, 5를 가질 수 있다.
  - 예를 들어  $N=16$ 이고  $K=5$ 일 때, 각 바구니에 공의 개수로 1, 2, 3, 4, 6를 가질 수 있다.
  - 예를 들어  $N=17$ 이고  $K=5$ 일 때, 각 바구니에 공의 개수로 1, 2, 3, 5, 6를 가질 수 있다.
- 항상 **정답은  $K-1$  혹은  $K$** 인 것을 알 수 있다.

JavaScript 탐욕법  
그리디 문제 풀이

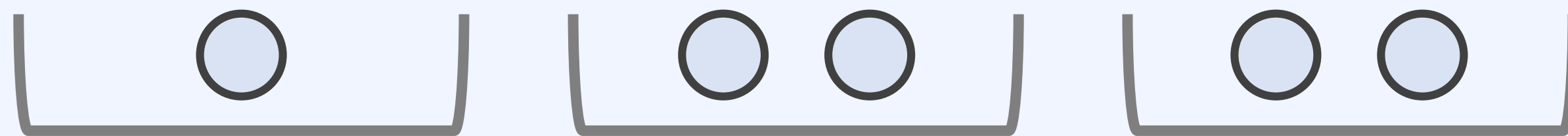
## 문제 해결 아이디어

JavaScript  
탐욕법  
그리디  
문제 풀이

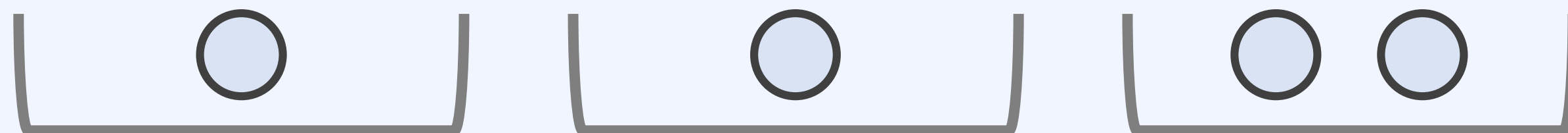
- $N$ 이 너무 작으면 목표를 만족할 수 없다. → 조건:  $1 + \dots + K \leq N$
- 가능한 예시)  $K = 3, N = 6$



- 불가능한 예시 1)  $K = 3, N = 5$



- 불가능한 예시 2)  $K = 3, N = 4$



JavaScript 탐욕법  
그리디 문제 풀이

## 문제 해결 아이디어

JavaScript  
탐욕법  
그리디  
문제 풀이

- $N$ 이 충분히 크면 목표를 만족할 수 있다.
- 가능한 예시)  $K = 4, N = 10 \rightarrow$  정답: 3



- 가능한 예시)  $K = 4, N = 11 \rightarrow$  정답: 4



- 가능한 예시)  $K = 4, N = 12 \rightarrow$  정답: 4



## 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');

// 공의 개수 N과 바구니의 개수 K를 입력받기
let n = Number(input[0].split(' ')[0]);
let k = Number(input[0].split(' ')[1]);

let summary = 0; // 1부터 K까지의 합
for (let i = 1; i < k + 1; i++) {
    summary += i;
}
if (summary > n) { // 공의 개수가 부족한 경우
    console.log(-1);
}
else { // 공의 개수가 충분한 경우
    n -= summary;
    if (n % k == 0) console.log(k - 1); // K개에 각각 1씩 더하기
    else console.log(k);
}
```

JavaScript 탐욕법  
그리디 문제 풀이

## 혼자 힘으로 풀어보기

JavaScript  
탐욕법  
그리디  
문제 풀이

문제 제목: 회문

문제 난이도: ★★☆☆☆

문제 유형: 그리디

추천 풀이 시간: 40분

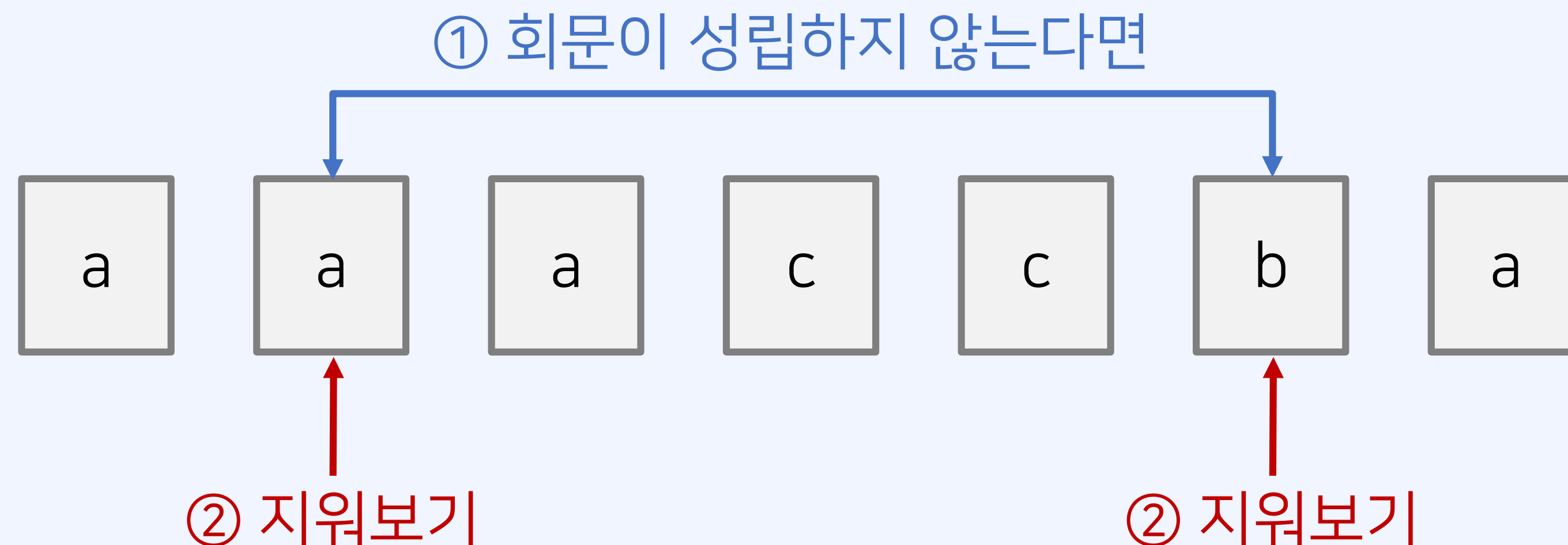


JavaScript 탐욕법  
그리디 문제 풀이

## 문제 해결 아이디어

JavaScript  
탐욕법  
그리디  
문제 풀이

- 유사회문: 한 문자를 삭제하여 회문으로 만들 수 있는 문자열
- 문자열의 앞에서부터 한 문자씩 확인하면서 회문이 성립하는지 확인한다.
- 만약 회문이 성립하지 않는 위치를 찾다면 다음의 과정으로 유사회문이 가능한지 여부를 판별한다.
  1. 해당 문자를 지웠을 때 유사회문이 될 수 있는지 확인한다.
  2. 대칭된 위치에 있는 문자를 지웠을 때 유사회문이 될 수 있는지 확인한다.



## 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');

function palindrome(x) {
    return x == x.split('').reverse().join('');
}

let testCases = Number(input[0]);
for (let tc = 1; tc <= testCases; tc++) { // 문자열을 하나씩 입력받아 처리
    let data = input[tc];
    if (palindrome(data)) console.log(0); // 회문인 경우
    else { // 회문이 아닌 경우, 유사 회문인지 검사
        let found = false;
        let n = data.length; // 문자열의 길이
        for (let i = 0; i < parseInt(n / 2); i++) {
            if (data[i] != data[n - i - 1]) { // 대칭이 아닌 인덱스를 찾은 경우
                // 앞쪽에 있는 해당 원소를 제거해 본 뒤에 회문 검사
                if (palindrome(data.slice(0, i) + data.slice(i + 1, n))) found = true;
                // 뒤쪽에 있는 해당 원소를 제거해 본 뒤에 회문 검사
                if (palindrome(data.slice(0, n - i - 1) + data.slice(n - i, n))) found = true;
                break;
            }
        }
        if (found) console.log(1); // 유사 회문인 경우
        else console.log(2); // 회문도 유사 회문도 아닌 경우
    }
}
```

JavaScript 탐욕법  
그리디 문제 풀이

## 혼자 힘으로 풀어보기

JavaScript  
탐욕법  
그리디  
문제 풀이

문제 제목: 박스 채우기

문제 난이도: ★★☆☆☆

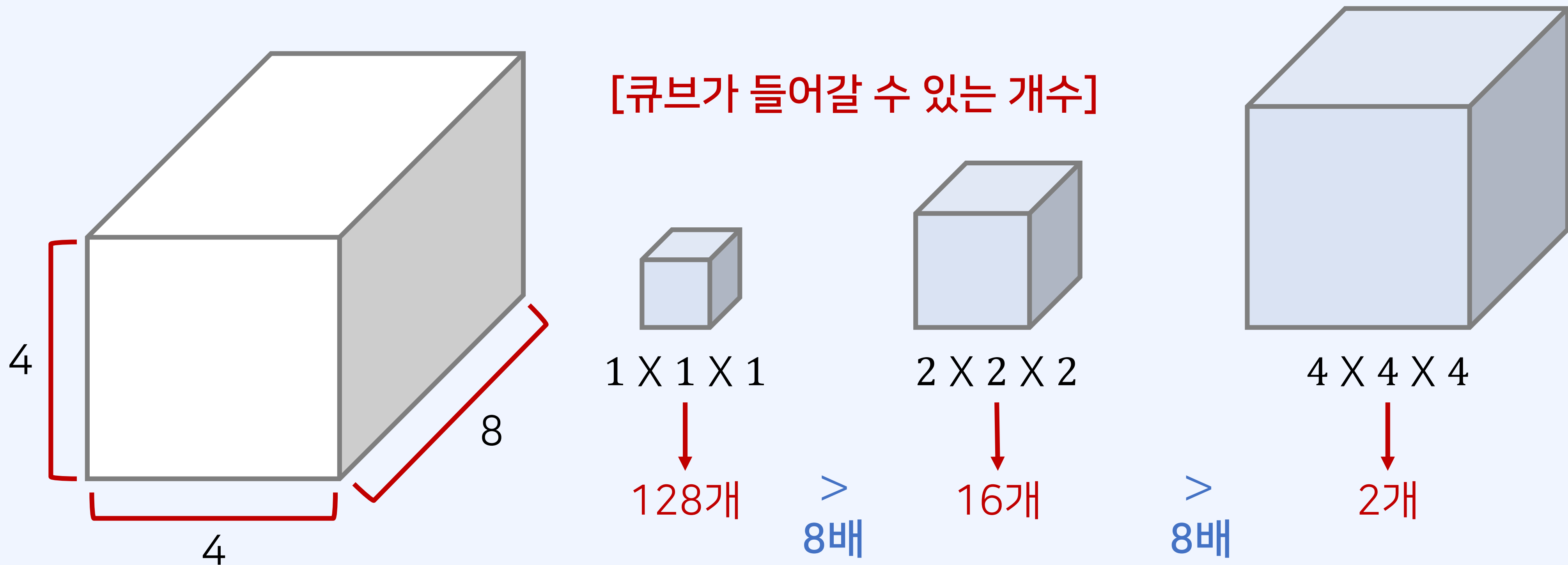
문제 유형: 그리디

추천 풀이 시간: 50분

- 큐브의 길이(length), 너비(width), 높이(height)는 항상 **2의 제곱 형태**를 보인다.
- **[문제 해결 아이디어]** 큰 큐브는 항상 자기보다 작은 큐브로 채울 수 있다.
- 결과적으로, 큰 큐브부터 차근차근 채워 나가면 해결할 수 있다.
- 그리디 알고리즘 유형의 대표적인 예시인 **[거스름 돈]** 문제와 유사하다.

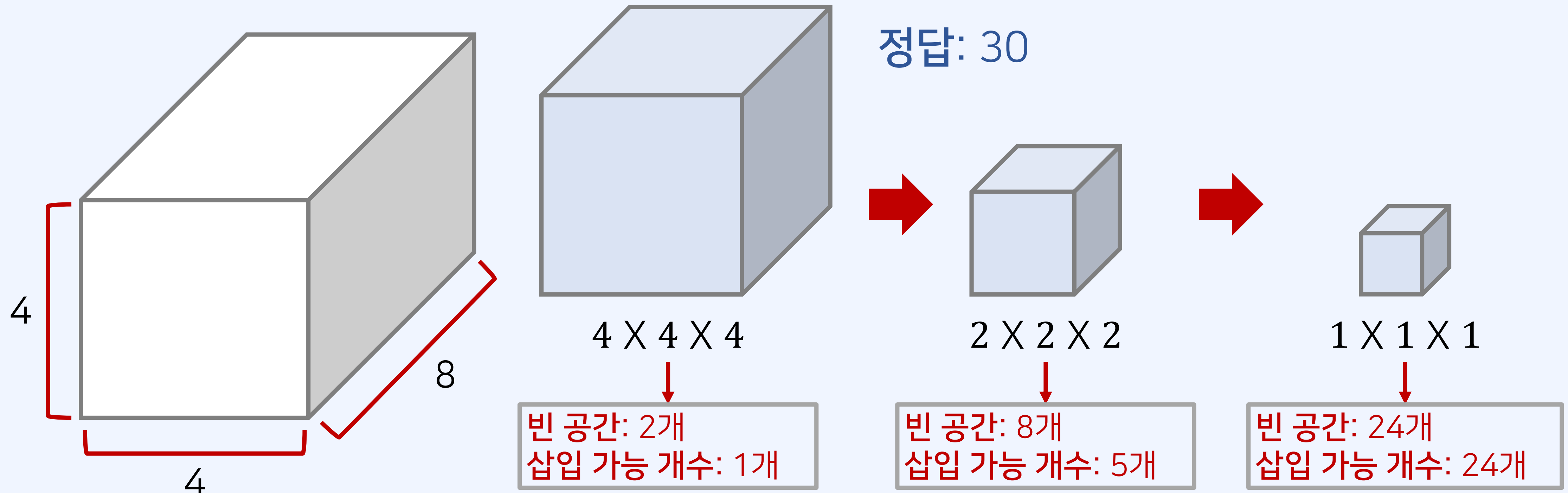
**[문제 해결 아이디어]** 큰 큐브는 항상 자기보다 작은 큐브로 채울 수 있다.

- 각 큐브를 몇 개 넣을 수 있는지 계산하기 위해  $(2^i \times 2^i \times 2^i)$  격자로 전체 박스를 나누어 확인한다.



**[문제 해결 아이디어]** 큰 큐브는 항상 자기보다 작은 큐브로 채울 수 있다.

- **해결 방법:** 최대한 큰 큐브부터 박스에 넣고, 다음 크기의 큐브도 박스에 넣을 수 있는지 단계적으로 확인한다.
- **[예시]** 가지고 있는 큐브의 개수가 (1 X 1 X 1) 32개, (2 X 2 X 2) 5개, (4 X 4 X 4) 1개라면?



## 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');

// x보다 작거나 같으면서 가장 가까운 2^i를 찾는 함수
function nearestSquare(x) {
    let i = 1;
    while ((2 ** i) <= x) i += 1;
    return i - 1;
}

let length = Number(input[0].split(' ')[0]);
let width = Number(input[0].split(' ')[1]);
let height = Number(input[0].split(' ')[2]);
let cubes = Array(20).fill(0);

let n = Number(input[1]);
for (let i = 2; i <= n + 1; i++) {
    let a = Number(input[i].split(' ')[0]);
    let b = Number(input[i].split(' ')[1]);
    cubes[a] = b;
}
```

## JavaScript 탐욕법 그리디 문제 풀이

## 정답 코드 예시

## JavaScript 탐욕법 그리디 문제 풀이

```
let size = 19;
size = nearestSquare(length);
size = Math.min(size, nearestSquare(width));
size = Math.min(size, nearestSquare(height));

let res = 0;
let used = 0;

for (let i = size; i >= 0; i--) {
    used *= 8; // 채널, 너비, 높이가 2씩 줄었으므로 큐브의 개수는 8배 증가
    cur = (2 ** i); // 현재의 정육면체 큐브의 길이
    // 채워넣어야 할 큐브의 개수 계산
    let required = parseInt(length / cur)
        * parseInt(width / cur)
        * parseInt(height / cur)
        - used;

    let usage = Math.min(required, cubes[i]); // 이번 단계에서 넣을 수 있는 큐브의 개수
    res += usage;
    used += usage;
}

if (used == length * width * height) console.log(res); // 박스를 가득 채운 경우
else console.log(-1); // 박스를 가득 채우지 못한 경우
```