

# JavaScript

## 다이나믹 프로그래밍

### 다이나믹 프로그래밍 문제 풀이

다이나믹 프로그래밍 이해하기 | 코딩 테스트에서 자주 등장하는 다이나믹 프로그래밍 이해하기

강사 나동빈

# JavaScript

## 다이나믹 프로그래밍

다이나믹 프로그래밍 문제 풀이

## JavaScript 동적 계획법 **혼자 힘으로 풀어보기** 다이나믹 프로그래밍

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

문제 제목: 병사 배치하기

문제 난이도: ★★☆☆☆

문제 유형: 다이나믹 프로그래밍

추천 풀이 시간: 40분

## JavaScript 동적 계획법 문제 해결 아이디어

### 다이나믹 프로그래밍

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

- $N$ 명의 병사가 무작위로 나열되어 있다.
- 각 병사는 특정한 값의 전투력을 보유하고 있다.
- 병사를 배치할 때는 전투력이 높은 병사가 앞쪽에 오도록 내림차순으로 배치를 하고자 한다.
- 다시 말해 앞쪽에 있는 병사의 전투력이 항상 뒤쪽에 있는 병사보다 높아야 합니다.
- 또한 배치 과정에서는 특정한 위치에 있는 병사를 옆외시키는 방법을 이용한다.
- 결과적으로 남아 있는 병사의 수가 최대가 되도록 하는 것이 목표다.

## JavaScript 동적 계획법 문제 해결 아이디어

다이나믹 프로그래밍

- 예를 들어,  $N = 7$ 일 때 나열된 병사들의 전투력이 다음과 같다고 가정하자.

병사 번호	1	2	3	4	5	6	7
전투력	15	11	4	8	5	2	4

- 이때 3번 병사와 6번 병사를 제외시키면, 다음과 같이 남아 있는 병사의 수가 내림차순의 형태가 되며 5명이 된다. 이는 남아 있는 병사의 수가 최대가 되도록 하는 방법이다.

병사 번호	1	2	4	5	7
전투력	15	11	8	5	4

- 병사에 대한 정보가 주어졌을 때, 남아 있는 병사의 수가 최대가 되도록 하기 위해서 제외시켜야 하는 병사의 수를 출력하는 프로그램을 작성하여라.

## JavaScript 동적 계획법 문제 해결 아이디어

### 다이나믹 프로그래밍

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

- 예를 들어 하나의 수열  $array = \{4, 2, 5, 8, 4, 11, 15\}$ 이 있다고 하자.
  - 이 수열의 가장 긴 증가하는 부분 수열은  $\{4, 5, 8, 11, 15\}$ 이다.
- 이 문제의 기본 아이디어는 가장 긴 증가하는 부분 수열(Longest Increasing Subsequence, LIS)로 알려진 전형적인 다이나믹 프로그래밍 문제의 아이디어와 같다.
- 본 문제는 가장 긴 감소하는 부분 수열을 찾는 문제로 치환할 수 있으므로, LIS 알고리즘을 조금 수정하여 적용함으로써 정답을 도출할 수 있다.

## JavaScript 동적 계획법 문제 해결 아이디어

### 다이나믹 프로그래밍

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

- 가장 긴 증가하는 부분 수열(LIS) 알고리즘을 확인해 보자.
- $d[i] = arr[i]$ 를 마지막 원소로 가지는 부분 수열의 최대 길이
- 점화식은 다음과 같다.
- 모든  $0 \leq j < i$ 에 대하여,  $d[i] = \max(d[i], d[j] + 1)$  if  $arr[j] < arr[i]$

# JavaScript 동적 계획법 문제 해결 아이디어

다이나믹 프로그래밍

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

- 모든  $0 \leq j < i$ 에 대하여,  $d[i] = \max(d[i], d[j] + 1)$  if  $arr[j] < arr[i]$

(초기 상태)  $i = 0$

4	2	5	8	4	11	15
---	---	---	---	---	----	----

1	1	1	1	1	1	1
---	---	---	---	---	---	---

$i = 1$

1	1	1	1	1	1	1
---	---	---	---	---	---	---

$i = 2$

1	1	2	1	1	1	1
---	---	---	---	---	---	---

$i = 3$

1	1	2	3	1	1	1
---	---	---	---	---	---	---

$i = 4$

1	1	2	3	2	1	1
---	---	---	---	---	---	---

$i = 5$

1	1	2	3	2	4	1
---	---	---	---	---	---	---

$i = 6$

1	1	2	3	2	4	5
---	---	---	---	---	---	---



## JavaScript 동적 계획법 문제 해결 아이디어

다이나믹 프로그래밍

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

- 가장 먼저 입력 받은 병사 정보의 순서를 뒤집는다.
- 가장 긴 증가하는 부분 수열 (LIS) 알고리즘을 수행하여 정답을 도출할 수 있다.

## JavaScript 동적 계획법

### 정답 코드 예시

#### 다이나믹 프로그래밍

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

```
// readline 모듈보다는 fs를 이용해 파일 전체를 읽기
let file = require('fs').readFileSync('/dev/stdin');
let input = file.toString().split('\n');

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

// 순서를 뒤집어 '최장 증가 부분 수열' 문제로 변환
arr.reverse();

// 다이나믹 프로그래밍을 위한 1차원 DP 테이블 초기화
dp = new Array(n).fill(1);

// 가장 긴 증가하는 부분 수열(LIS) 알고리즘 수행
for (let i = 1; i < n; i++) {
  for (let j = 0; j < i; j++) {
    if (arr[j] < arr[i]) {
      dp[i] = Math.max(dp[i], dp[j] + 1);
    }
  }
}

// 열외해야 하는 병사의 최소 수를 출력
console.log(n - Math.max(... dp));
```

## JavaScript 동적 계획법 **혼자 힘으로 풀어보기** 다이나믹 프로그래밍

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

문제 제목: 1로 만들기

문제 난이도: ★★☆☆☆

문제 유형: 다이나믹 프로그래밍

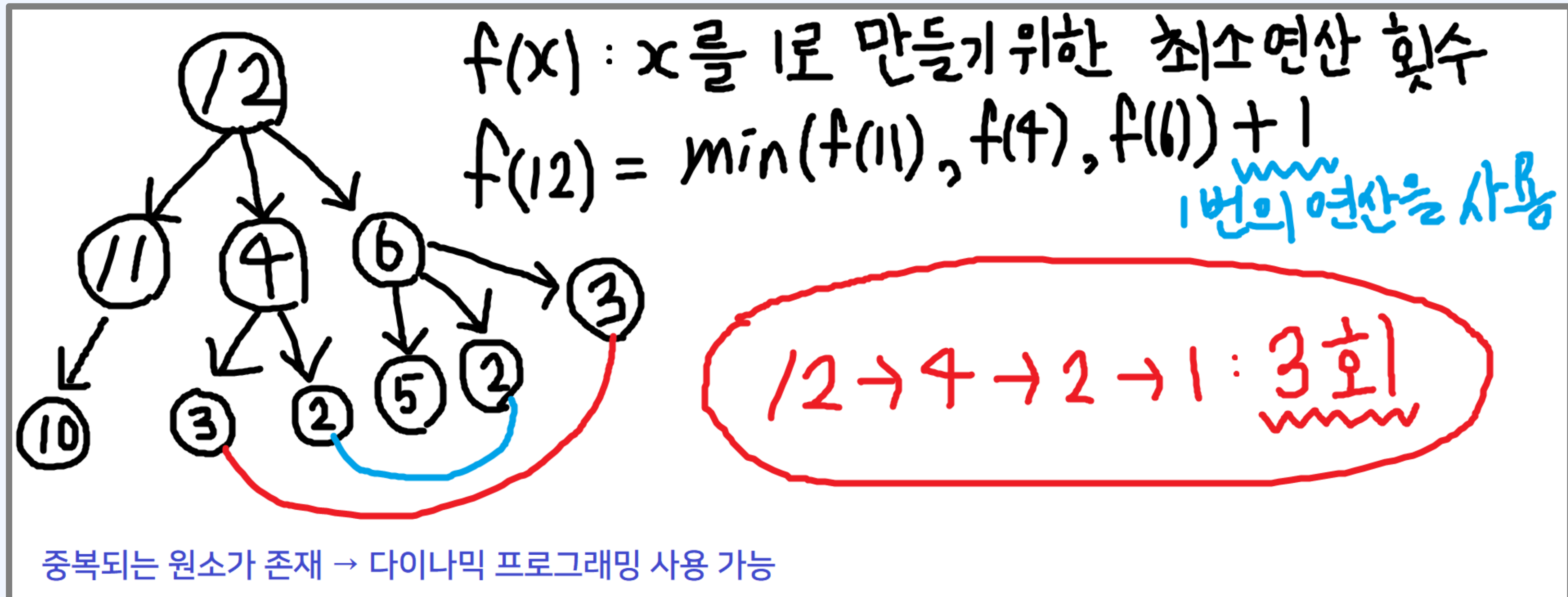
추천 풀이 시간: 30분

# JavaScript 동적 계획법 문제 해결 아이디어

## 다이나믹 프로그래밍

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

- 점화식을 도출하기 위해 몇 가지 예시를 고려해 볼 수 있다.



## JavaScript 동적 계획법 다이나믹 프로그래밍

### 정답 코드 예시

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

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

let n = Number(input[0]);
let d = new Array(n + 1).fill(0);

for (let x = 2; x <= n; x++) {
  d[x] = d[x - 1]; // 1을 빼기
  if (x % 2 == 0) {
    d[x] = Math.min(d[x], d[parseInt(x / 2)]); // 2로 나누기
  }
  if (x % 3 == 0) {
    d[x] = Math.min(d[x], d[parseInt(x / 3)]); // 3으로 나누기
  }
  d[x]++;
}

console.log(d[n]);
```