

# 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: 피보나치 함수

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 직접 코드를 실행하며 확인할 수 있다.

```
// 0과 1이 몇 번씩 출력되는지 확인하는 소스코드
let zero = 0;
let one = 0;

function fibonacci(n) {
  if (n == 0) {
    zero++;
    return 0;
  } else if (n == 1) {
    one++;
    return 1;
  } else {
    return fibonacci(n-1) + fibonacci(n-2);
  }
}

let n = 10; // 이 값을 바꾸어 가면서 확인하기
console.log(n, fibonacci(n), zero, one);
```

## JavaScript 동적 계획법

### 정답 코드 예시

#### 다이나믹 프로그래밍

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

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

// 앞서 계산된 결과를 저장하기 위한 DP 테이블 초기화
d = new Array(100).fill(0);
d[0] = 0;
d[1] = 1;

// 피보나치 함수(Fibonacci Function) 반복문으로 구현(보텀업)
for (let i = 2; i <= 40; i++) {
    d[i] = d[i - 1] + d[i - 2];
}

let testCases = Number(input[0]);
for (let t = 1; t <= testCases; t++) {
    // N번째 피보나치 수에 대하여
    let n = Number(input[t]);
    // 0의 개수와 1의 개수 출력
    if (n == 0) console.log(1, 0);
    else console.log(d[n - 1], d[n]);
}
```

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

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

문제 제목: 01타일

문제 난이도: ★★☆☆☆

문제 유형: 다이나믹 프로그래밍

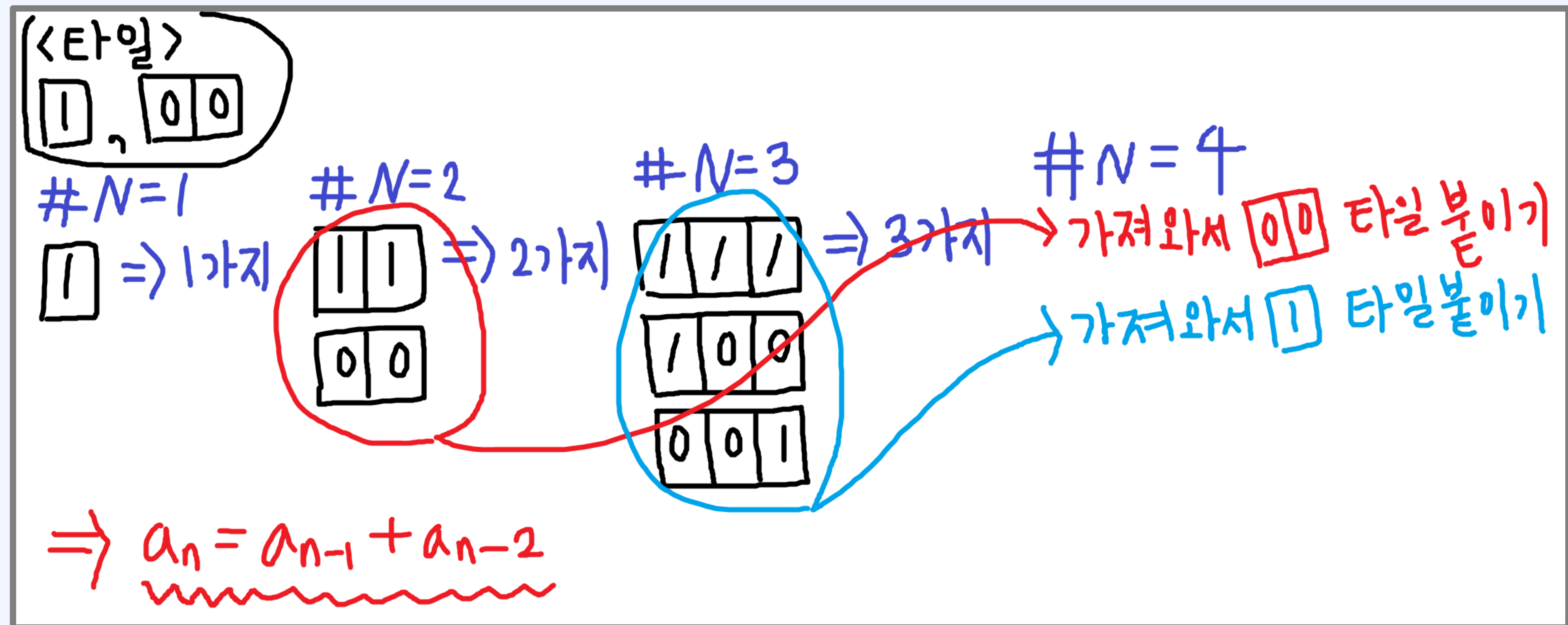
추천 풀이 시간: 30분

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

let n = Number(input[0]); // N 입력받기

// 앞서 계산된 결과를 저장하기 위한 DP 테이블 초기화
d = new Array(1000001).fill(0);
d[1] = 1;
d[2] = 2;

// 다이나믹 프로그래밍(Dynamic Programming) 진행 (보텀업)
for (let i = 3; i <= n; i++) {
    // d[i] = (i) 1을 붙이는 경우 + (i-1) 00을 붙이는 경우
    // 참고로 11을 붙이는 경우는 포함하지 않는데, (i)에 포함되기 때문이다.
    d[i] = (d[i - 1] + d[i - 2]) % 15746;
    // 값이 매우 커질 수 있기 때문에 나머지를 구하는데, 더할 때마다 나머지만 저장하면 된다.
    // 예를 들어 (40 + 25 + 15) % 7 = ((40 % 7 + 25) % 7 + 15) % 7이다.
}

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



## JavaScript 동적 계획법 다이나믹 프로그래밍

# 혼자 힘으로 풀어보기

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

문제 제목: 포도주 시식

문제 난이도: ★★☆☆☆

문제 유형: 다이나믹 프로그래밍

추천 풀이 시간: 40분

# JavaScript 동적 계획법 문제 해결 아이디어

## 다이나믹 프로그래밍

- 점화식을 도출하기 위해 몇 가지 예시를 고려해 볼 수 있다.

6 10 13 9 8 1

문제: 제일 많이 마시기 (연속 3잔은 X)

	6	10	13	9	8	1	
1)	✓	✓		✓	✓		⇒ 33
2)	✓		✓	✓			⇒ 28
3)		✓	✓		✓		⇒ 31

## JavaScript 동적 계획법 문제 해결 아이디어

다이나믹 프로그래밍

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

- 점화식을 도출하기 위해 몇 가지 예시를 고려해 볼 수 있다.
- 01타일 문제와 유사한 아이디어로 점화식을 도출할 수 있다.

$$dp[i] = \text{\underline{i}번째 포도주까지의 최적의 해 (최대값)}$$

최대값 선택

$$\left\{ \begin{array}{l} i\text{번째만 마시기} : dp[i-1] \\ i\text{번째까지 마시고, } \left\{ \begin{array}{l} arr[i] + dp[i-2] \\ arr[i] + arr[i-1] + dp[i-3] \end{array} \right. \end{array} \right.$$

## JavaScript 동적 계획법

### 정답 코드 예시

#### 다이나믹 프로그래밍

JavaScript  
동적 계획법  
다이나믹  
프로그래밍

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

let n = Number(input[0]); // 포도주의 개수(N)
let arr = [];
for (let i = 1; i <= n; i++) {
    arr.push(Number(input[i]));
}

let dp = new Array(n).fill(0);
dp[0] = arr[0];
dp[1] = arr[0] + arr[1];
dp[2] = Math.max(arr[0] + arr[1], arr[0] + arr[2], arr[1] + arr[2]);

// 점화식에 따라서 DP 테이블 갱신
for (let i = 3; i < n; i++) {
    dp[i] = dp[i - 1];
    dp[i] = Math.max(dp[i], arr[i] + dp[i - 2]);
    dp[i] = Math.max(dp[i], arr[i] + arr[i - 1] + dp[i - 3]);
}
console.log(dp[n - 1]);
```