

DP

최백준 choi@startlink.io

1로 만들기

<https://www.acmicpc.net/problem/1463>

- 세준이는 어떤 정수 N 에 다음과 같은 연산중 하나를 할 수 있다.
 1. N 이 3으로 나누어 떨어지면, 3으로 나눈다.
 2. N 이 2로 나누어 떨어지면, 2로 나눈다.
 3. 1을 뺀다.
- 세준이는 어떤 정수 N 에 위와 같은 연산을 선택해서 1을 만드려고 한다. 연산을 사용하는 횟수의 최소값을 출력하시오.

1로 만들기

<https://www.acmicpc.net/problem/1463>

- $D[i] = i$ 를 1로 만드는데 필요한 최소 연산 횟수
- i 에게 가능한 경우를 생각해보자
 1. i 가 3으로 나누어 떨어졌을 때, 3으로 나누는 경우
 2. i 가 2로 나누어 떨어졌을 때, 2로 나누는 경우
 3. i 에서 1을 빼는 경우

1로 만들기

<https://www.acmicpc.net/problem/1463>

- $D[i]$ = i 를 1로 만드는데 필요한 최소 연산 횟수
- i 에게 가능한 경우를 생각해보자
 1. i 가 3으로 나누어 떨어졌을 때, 3으로 나누는 경우
 - $D[i/3] + 1$
 2. i 가 2로 나누어 떨어졌을 때, 2로 나누는 경우
 - $D[i/2] + 1$
 3. i 에서 1을 빼는 경우
 - $D[i-1] + 1$

1로 만들기

<https://www.acmicpc.net/problem/1463>

- $D[i]$ = i 를 1로 만드는데 필요한 최소 연산 횟수
- i 에게 가능한 경우를 생각해보자
 1. i 가 3으로 나누어 떨어졌을 때, 3으로 나누는 경우
 - $D[i/3] + 1$
 2. i 가 2로 나누어 떨어졌을 때, 2로 나누는 경우
 - $D[i/2] + 1$
 3. i 에서 1을 빼는 경우
 - $D[i-1] + 1$
- 세 값중의 최소값이 들어가게 된다.

1로 만들기

<https://www.acmicpc.net/problem/1463>

```
int go(int n) {  
    if (n == 1) return 0;  
    if (d[n] > 0) return d[n];  
    d[n] = go(n-1) + 1;  
    if (n%2 == 0) {  
        int temp = go(n/2) + 1;  
        if (d[n] > temp) d[n] = temp;  
    }  
    if (n%3 == 0) {  
        int temp = go(n/3) + 1;  
        if (d[n] > temp) d[n] = temp;  
    }  
    return d[n];  
}
```

1로 만들기

<https://www.acmicpc.net/problem/1463>

```
d[1] = 0;
for (int i=2; i<=n; i++) {
    d[i] = d[i-1] + 1;
    if (i%2 == 0 && d[i] > d[i/2] + 1) {
        d[i] = d[i/2] + 1;
    }
    if (i%3 == 0 && d[i] > d[i/3] + 1) {
        d[i] = d[i/3] + 1;
    }
}
```

1로 만들기

<https://www.acmicpc.net/problem/1463>

- Top-Down 방식
- C: <https://gist.github.com/Baekjoon/a53dc4861bd9d081682c>
- C++: <https://gist.github.com/Baekjoon/63b659f985beb8f64ca7>
- Java: <https://gist.github.com/Baekjoon/7b675fe68d3c2abfef40>

1로 만들기

<https://www.acmicpc.net/problem/1463>

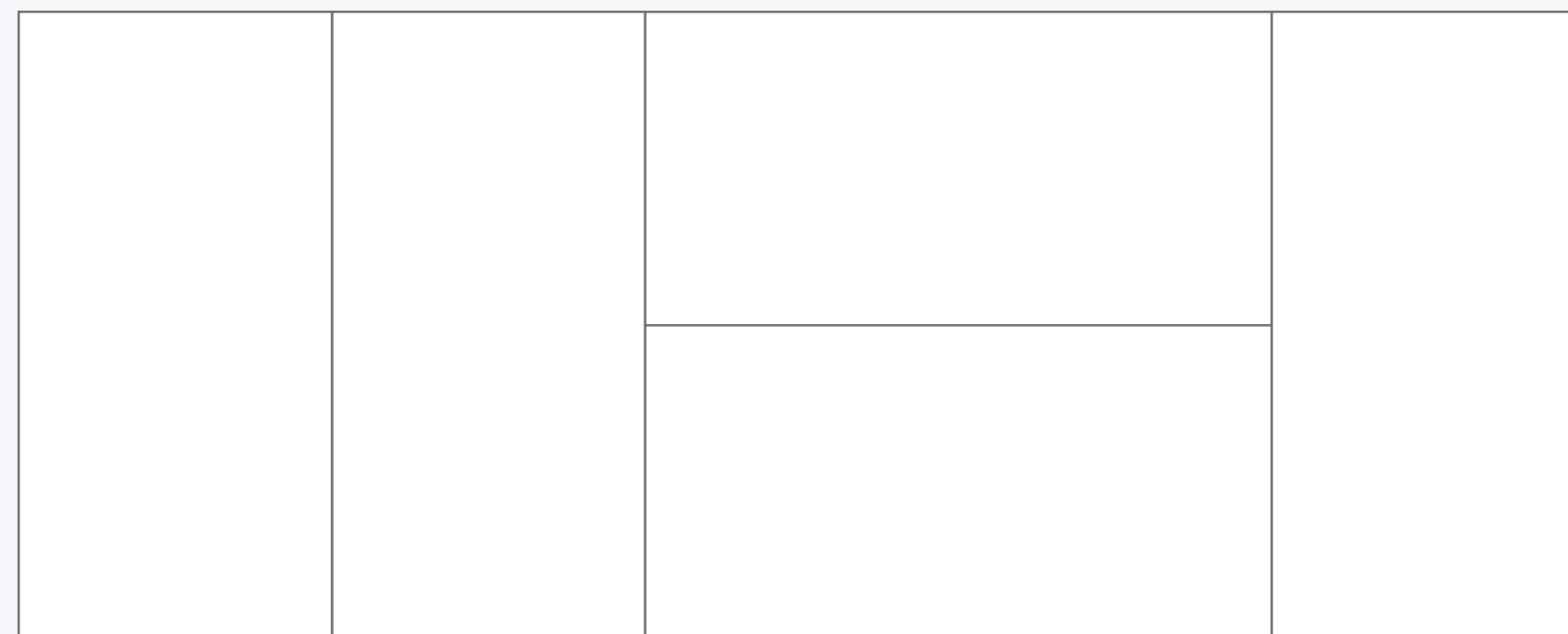
- Bottom-up 방식
- C: <https://gist.github.com/Baekjoon/30f4bb39cdc66f7f16c1>
- C++: <https://gist.github.com/Baekjoon/31e553ab3b371fe06384>
- Java: <https://gist.github.com/Baekjoon/0813d3bc5db11b9bb72d>

$2 \times n$ 타일링

10

<https://www.acmicpc.net/problem/11726>

- $2 \times n$ 직사각형을 1×2 , 2×1 타일로 채우는 방법의 수
- 아래 그림은 2×5 를 채우는 방법의 수
- $D[i] = 2 \times i$ 직사각형을 채우는 방법의 수

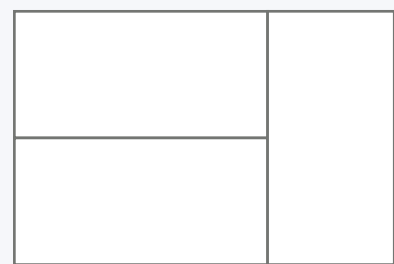
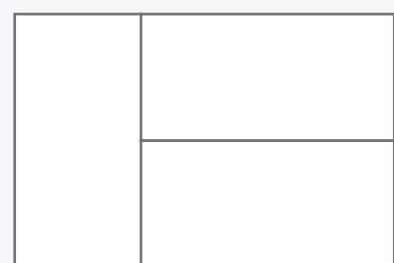
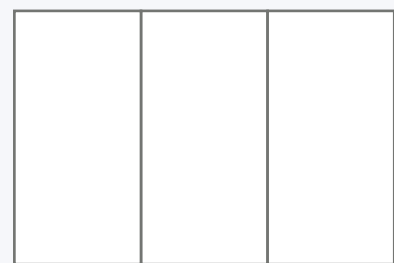


2×n 타일링

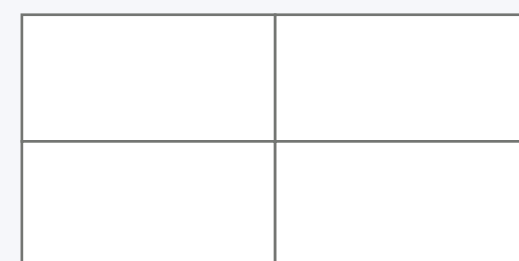
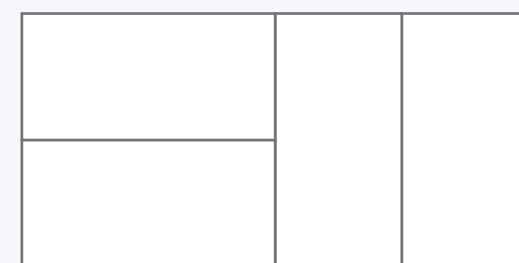
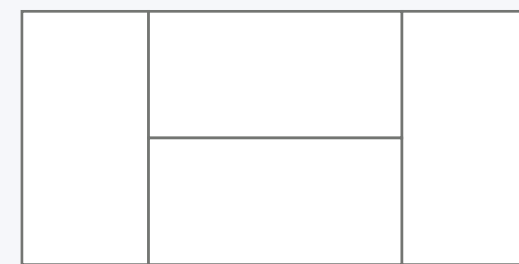
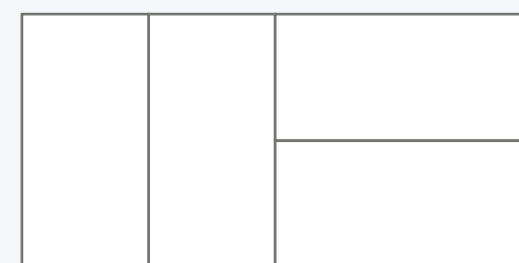
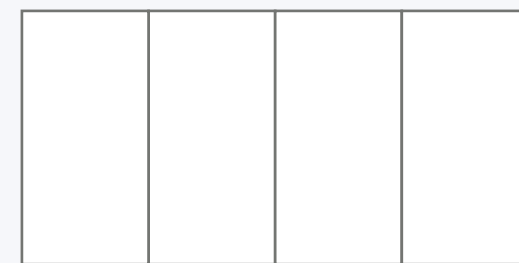
<https://www.acmicpc.net/problem/11726>

11

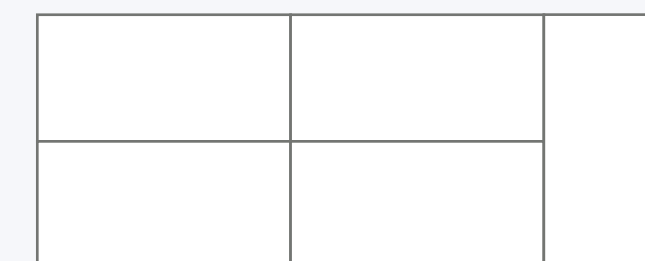
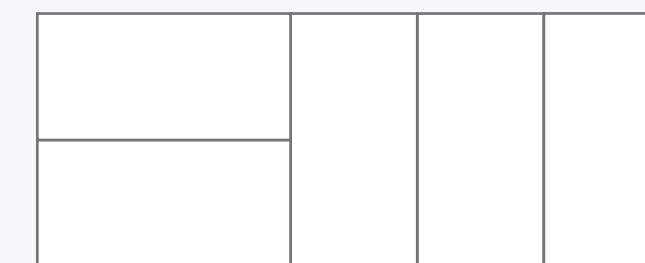
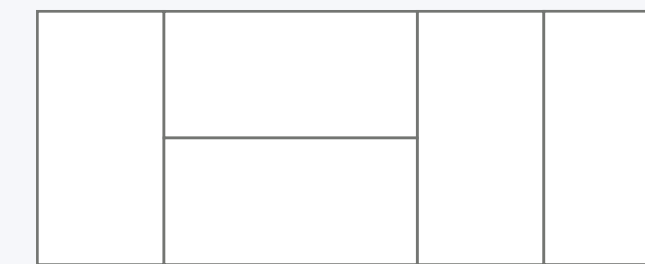
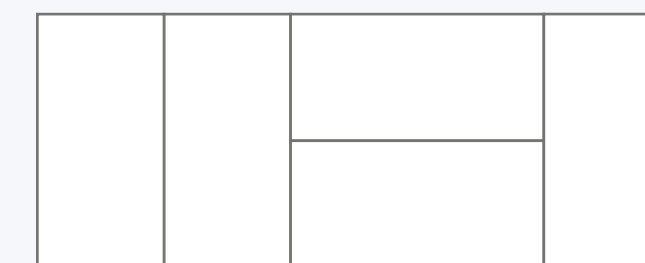
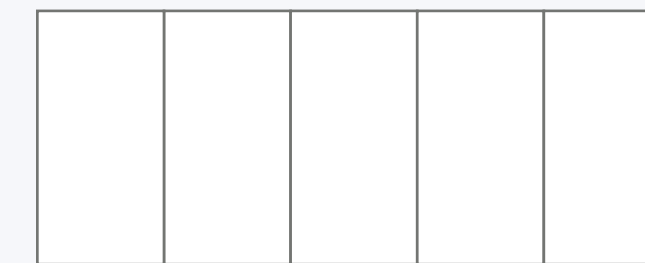
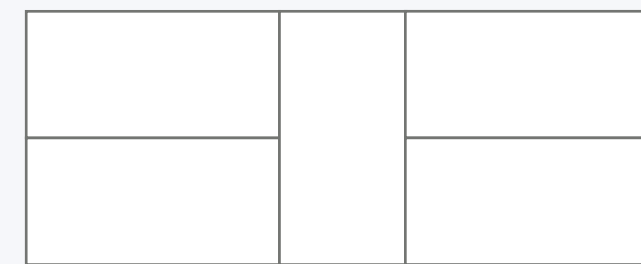
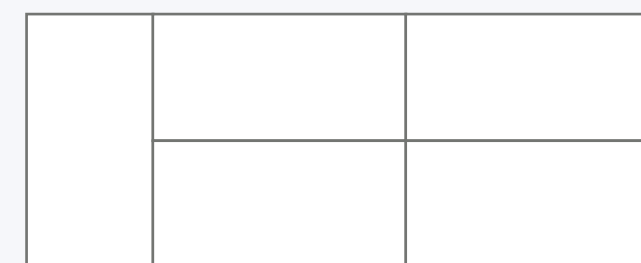
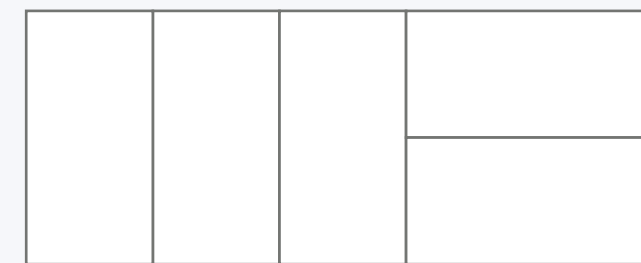
2×3



2×4



2×5



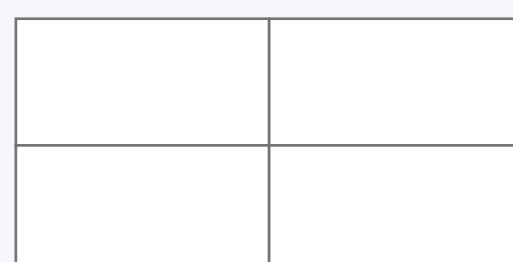
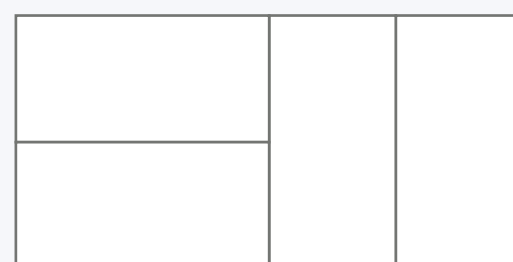
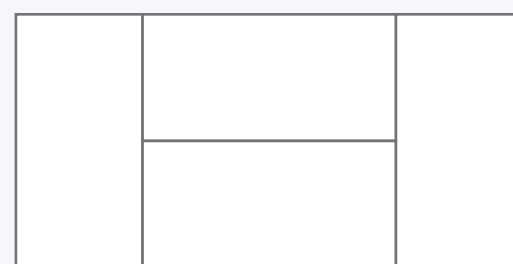
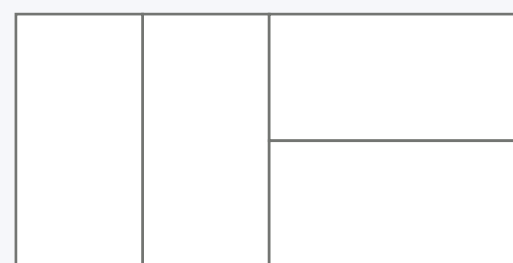
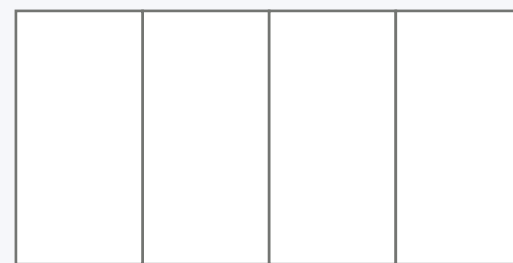
2×n 타일링

<https://www.acmicpc.net/problem/11726>

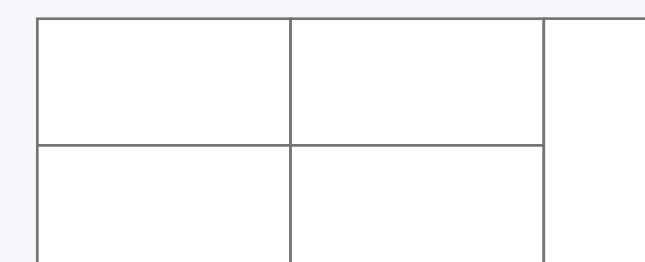
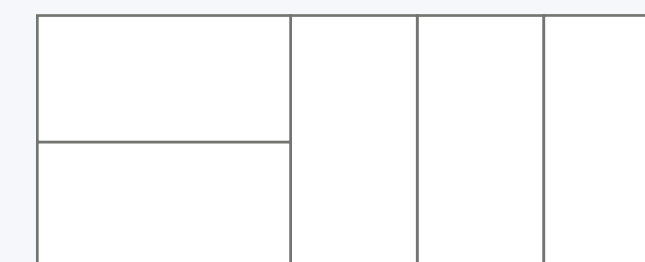
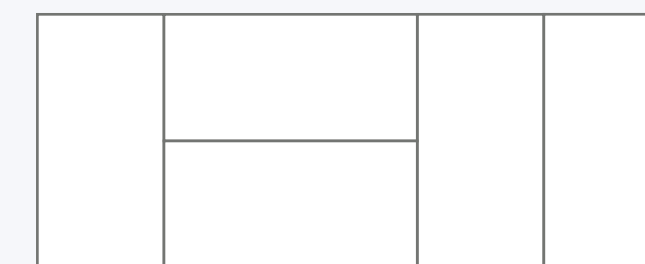
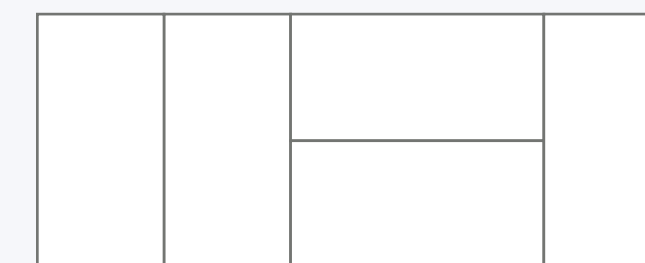
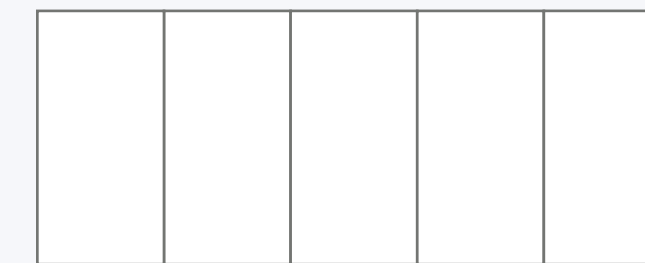
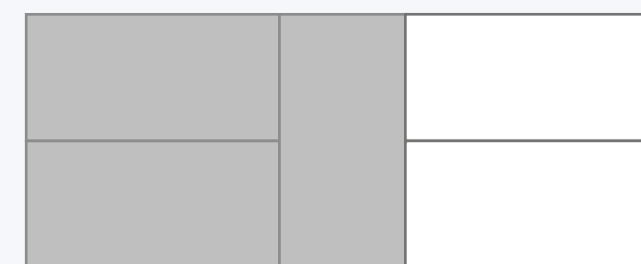
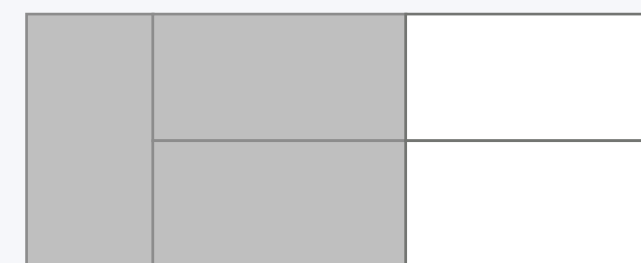
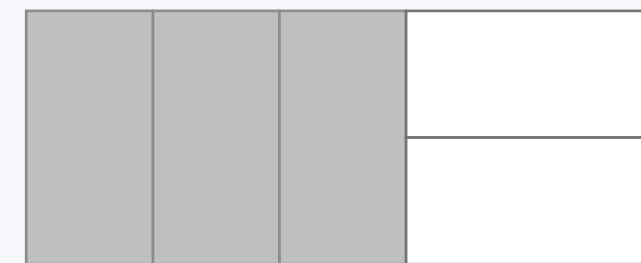
2×3



2×4



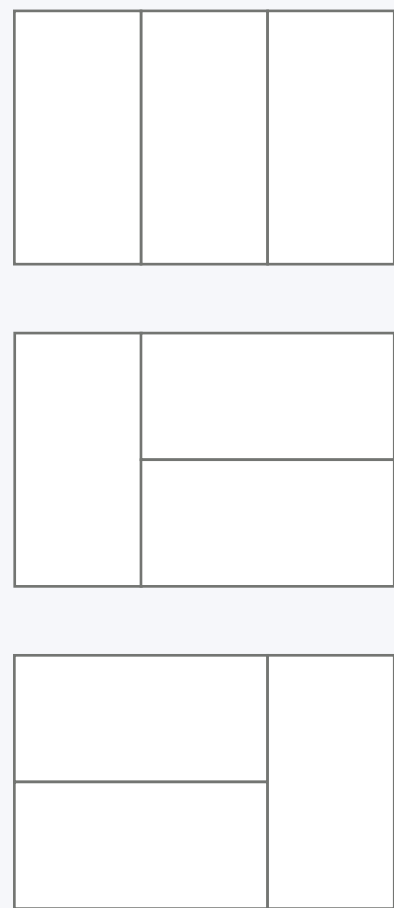
2×5



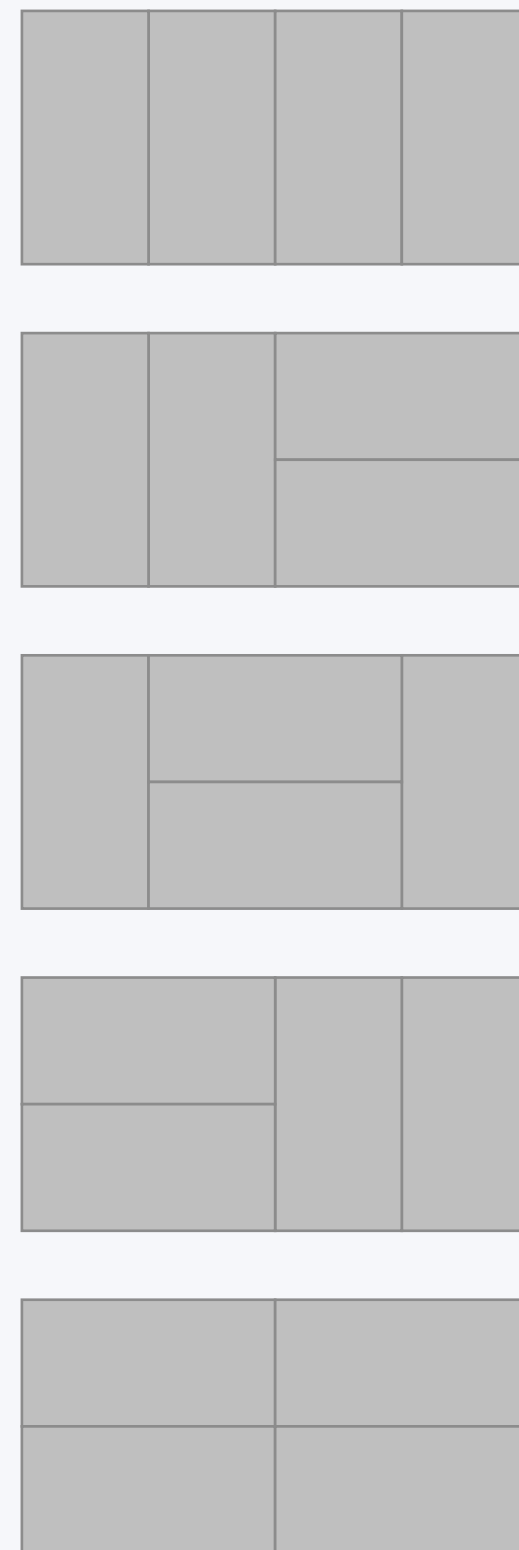
2×n 타일링

<https://www.acmicpc.net/problem/11726>

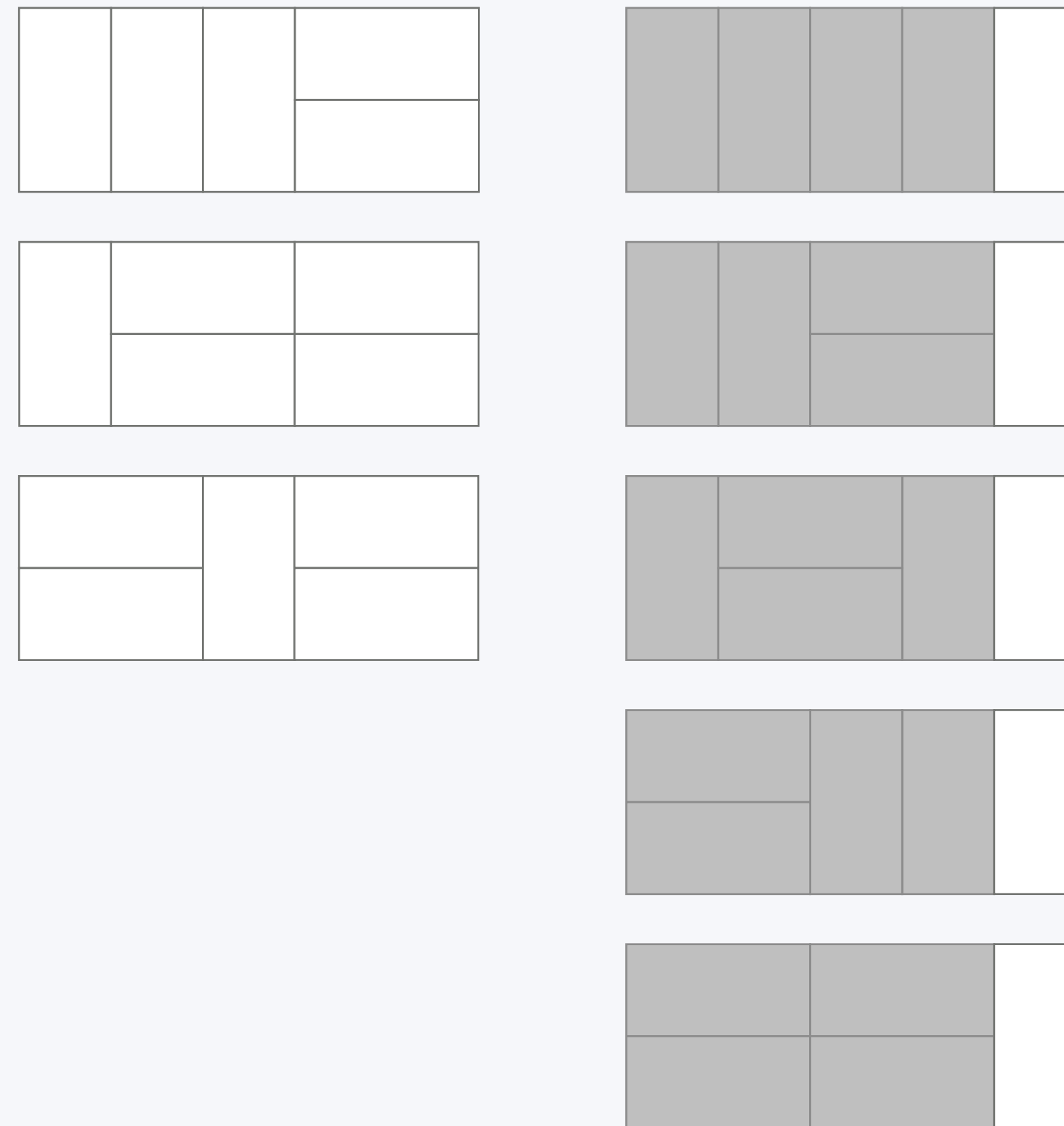
2×3



2×4



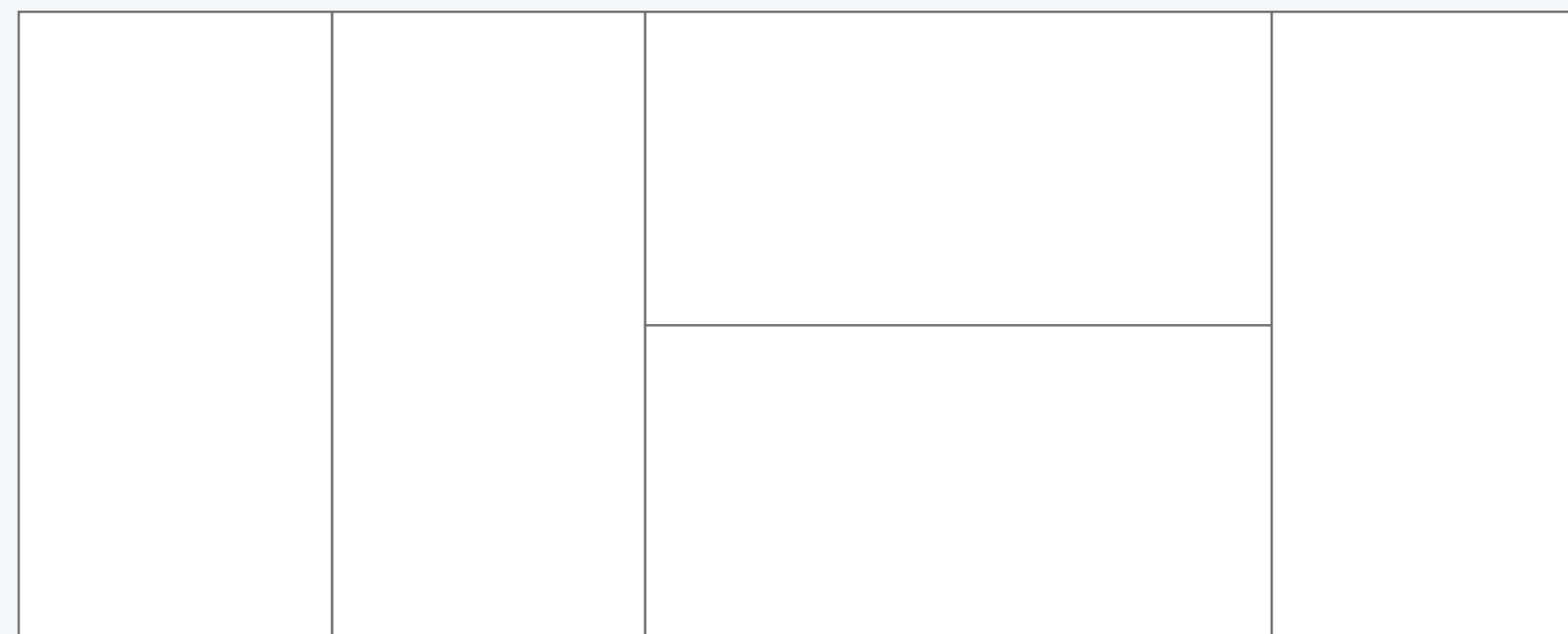
2×5



$2 \times n$ 타일링

<https://www.acmicpc.net/problem/11726>

- $2 \times n$ 직사각형을 1×2 , 2×1 타일로 채우는 방법의 수
- $D[i] = 2 \times i$ 직사각형을 채우는 방법의 수
- $D[i] = D[i-1] + D[i-2]$



2 × n 타일링

<https://www.acmicpc.net/problem/11726>

- C/C++: <https://gist.github.com/Baekjoon/3527f6fd4771f8c3e1>
- Java: <https://gist.github.com/Baekjoon/53f6e5ec06bfbafad977150df382cf55>

1, 2, 3 더하기

16

<https://www.acmicpc.net/problem/9095>

- 정수 n 을 1, 2, 3의 조합으로 나타내는 방법의 수를 구하는 문제
- $n = 4$
- $1+1+1+1$
- $1+1+2$
- $1+2+1$
- $2+1+1$
- $2+2$
- $1+3$
- $3+1$

1, 2, 3 더하기

<https://www.acmicpc.net/problem/9095>

- $D[i] = i$ 를 1, 2, 3의 조합으로 나타내는 방법의 수

1, 2, 3 더하기

18

<https://www.acmicpc.net/problem/9095>

- $D[i]$ = i 를 1, 2, 3의 조합으로 나타내는 방법의 수
- $D[i] = D[i-1] + D[i-2] + D[i-3]$

1, 2, 3 더하기

<https://www.acmicpc.net/problem/9095>

- C/C++: <https://gist.github.com/Baekjoon/6e4f9e363b3aaef733d1>
- Java: <https://gist.github.com/Baekjoon/e019984a7c7f1ac6bd32>

쉬운 계단 수

20

<https://www.acmicpc.net/problem/10844>

- 인접한 자리의 차이가 1이 나는 수를 계단 수라고 한다
- 예: 45656
- 길이가 N인 계단 수의 개수를 구하는 문제

쉬운 계단 수

<https://www.acmicpc.net/problem/10844>

- $D[i][j]$ = 길이가 i 이고 마지막 숫자가 j 인 계단 수의 개수
- $D[i][j] = D[i-1][j-1] + D[i-1][j+1]$

쉬운 계단 수

<https://www.acmicpc.net/problem/10844>

```
for (int i=1; i<=9; i++) d[1][i] = 1;
for (int i=2; i<=n; i++) {
    for (int j=0; j<=9; j++) {
        d[i][j] = 0;
        if (j-1 >= 0) d[i][j] += d[i-1][j-1];
        if (j+1 <= 9) d[i][j] += d[i-1][j+1];
        d[i][j] %= mod;
    }
}

long long ans = 0;
for (int i=0; i<=9; i++) ans += d[n][i];
ans %= mod;
```

쉬운 계단 수

<https://www.acmicpc.net/problem/10844>

- C/C++: <https://gist.github.com/Baekjoon/4d98f519afbcdd5d3d0f>
- Java: <https://gist.github.com/Baekjoon/7e4e12ce1b0aa740d5d1>

오르막 수

<https://www.acmicpc.net/problem/11057>

- 오르막 수는 수의 자리가 오름차순을 이루는 수를 말한다
- 인접한 수가 같아도 오름차순으로 친다
- 수의 길이 N이 주어졌을 때, 오르막 수의 개수를 구하는 문제
- 수는 0으로 시작할 수 있다
- 예: 1233345, 357, 8888888, 1555999

오르막 수

<https://www.acmicpc.net/problem/11057>

- $D[i][j]$ = 길이가 i 이고 마지막 숫자가 j 인 오르막 수의 개수
- $D[1][i] = 1$
- $D[i][j] += D[i-1][k] \ (0 \leq k \leq j)$

오르막 수

<https://www.acmicpc.net/problem/11057>

```
for (int i=0; i<=9; i++) d[1][i] = 1;
for (int i=2; i<=n; i++) {
    for (int j=0; j<=9; j++) {
        for (int k=0; k<=j; k++) {
            d[i][j] += d[i-1][k];
            d[i][j] %= mod;
        }
    }
}

long long ans = 0;
for (int i=0; i<10; i++) ans += d[n][i];
ans %= mod;
```

오르막 수

<https://www.acmicpc.net/problem/11057>

- C/C++: <https://gist.github.com/Baekjoon/3d7ae9472aa843dc3a48>
- Java: <https://gist.github.com/Baekjoon/264f68b19e93cc9b46aa>

이친수

<https://www.acmicpc.net/problem/2193>

- 0과 1로만 이루어진 수를 이진수라고 한다.
- 다음 조건을 만족하면 이친수라고 한다.
 1. 이친수는 0으로 시작하지 않는다.
 2. 이친수에서는 1이 두 번 연속으로 나타나지 않는다. 즉, 11을 부분 문자열로 갖지 않는다.
- N자리 이친수의 개수를 구하는 문제

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i][j]$ = i자리 이친수의 개수 중에서 j로 끝나는 것의 개수 ($j=0, 1$)
- 0으로 시작하지 않는다.
- $D[1][0] = 0$
- $D[1][1] = 1$

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i][j]$ = i자리 이친수의 개수 중에서 j로 끝나는 것의 개수 ($j=0, 1$)
- 가능한 경우
- 0으로 끝나는 경우
- 1로 끝나는 경우

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i][j]$ = i 자리 이친수의 개수 중에서 j 로 끝나는 것의 개수 ($j=0, 1$)
- 가능한 경우
- 0으로 끝나는 경우 ($D[i][0]$)
 - 앞에 0과 1이 올 수 있다
 - $D[i-1][0] + D[i-1][1]$
- 1로 끝나는 경우 ($D[i][1]$)
 - 앞에 1은 올 수 없다. 즉, 0만 올 수 있다.
 - $D[i-1][0]$

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i][j]$ = i자리 이친수의 개수 중에서 j로 끝나는 것의 개수 ($j=0, 1$)
- $D[i][0] = D[i-1][0] + D[i-1][1]$
- $D[i][1] = D[i-1][0]$

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i]$ = i 자리 이친수의 개수
- 가능한 경우
- 0으로 끝나는 경우
- 1로 끝나는 경우

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i]$ = i 자리 이친수의 개수
- 가능한 경우
- 0으로 끝나는 경우
 - 앞에 0과 1 모두 올 수 있다.
 - $D[i-1]$
- 1로 끝나는 경우
 - 앞에 0만 올 수 있다
 - 앞에 붙는 0을 세트로 생각해서 $i-2$ 자리에 01을 붙인다고 생각
 - $D[i-2]$

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i]$ = i자리 이친수의 개수
- $D[i] = D[i-1] + D[i-2]$

이친수

<https://www.acmicpc.net/problem/2193>

- C/C++: <https://gist.github.com/Baekjoon/49b2bfd22be42707bb88>
- Java: <https://gist.github.com/Baekjoon/7fbfd8d0963139d638de>

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

- 수열 A가 주어졌을 때, 가장 긴 증가하는 부분 수열을 구하는 문제
- 예시
- 수열 $A = \{10, 20, 10, 30, 20, 50\}$
- 가장 긴 증가하는 부분 수열 $A = \{10, 20, 10, 30, 20, 50\}$

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이
- $D[i]$ 은 $A[i]$ 이 반드시 포함되어야 한다.
- 가장 긴 부분 수열이 $A[?], A[?], \dots, A[j], A[i]$ 라고 했을 때, 겹치는 부분 문제를 찾아보자.

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이
- $D[i]$ 은 $A[i]$ 이 반드시 포함되어야 한다.
- 가장 긴 부분 수열이 $A[?], A[?], \dots, A[j], A[i]$ 라고 했을 때, 겹치는 부분 문제를 찾아보자.
- $A[?], A[?], \dots, A[j]$ 는 $D[j]$ 로 나타낼 수 있다. ($A[j]$ 을 마지막으로 하는 부분 수열이기 때문)
- 그럼 $A[j]$ 와 $A[i]$ 간의 관계를 생각해보자.

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이
- $D[i]$ 은 $A[i]$ 이 반드시 포함되어야 한다.
- 가장 긴 부분 수열이 $A[?], A[?], \dots, A[j], A[i]$ 라고 했을 때, 겹치는 부분 문제를 찾아보자.
- $A[?], A[?], \dots, A[j]$ 는 $D[j]$ 로 나타낼 수 있다. ($A[j]$ 을 마지막으로 하는 부분 수열이기 때문)
- 그럼 $A[j]$ 와 $A[i]$ 간의 관계를 생각해보자.
- $A[j] < A[i]$ 가 되어야 한다. (증가하는 부분 수열이 되어야 하기 때문)

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이
- $D[5]$ 를 나타낸 그림

A[1]	A[2]	A[3]	A[4]	A[5]
10	20	10	30	20



$A[5]$ 를 마지막으로 하는 증가하는 부분 수열

가장 긴 증가하는 부분 수열

42

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

A[1]	A[2]	A[3]	A[4]	A[5]
10	20	10	30	20



A[1]
10

A[1]	A[2]
10	20

A[1]	A[2]	A[3]
10	20	10

A[1]	A[2]	A[3]	A[4]
10	20	10	30

가장 긴 증가하는 부분 수열

43

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

A[1]	A[2]	A[3]	A[4]	A[5]
10	20	10	30	20



A[1]	10	$D[1] = 1$	A[5]	20	$10 < 20$
------	----	------------	------	----	-----------



A[1]	A[2]	10	20	$D[2] = 2$	A[5]	20	$20 == 20$
------	------	----	----	------------	------	----	------------



A[1]	A[2]	A[3]	10	20	10	$D[3] = 1$	A[5]	20	$10 < 20$
------	------	------	----	----	----	------------	------	----	-----------



A[1]	A[2]	A[3]	A[4]	10	20	10	30	$D[4] = 3$	A[5]	20	$30 > 20$
------	------	------	------	----	----	----	----	------------	------	----	-----------



가장 긴 증가하는 부분 수열

44

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

A[1]	A[2]	A[3]	A[4]	A[5]
10	20	10	30	20



$D[5] = 2$

A[1]	A[5]
10	20

$D[1] = 1$ $10 < 20$ ○

A[1]	A[2]	A[5]
10	20	20

$D[2] = 2$ $20 == 20$ ✗

A[1]	A[2]	A[3]	A[5]
10	20	10	20

$D[3] = 1$ $10 < 20$ ○

A[1]	A[2]	A[3]	A[4]	A[5]
10	20	10	30	20

$D[4] = 3$ $30 > 20$ ✗

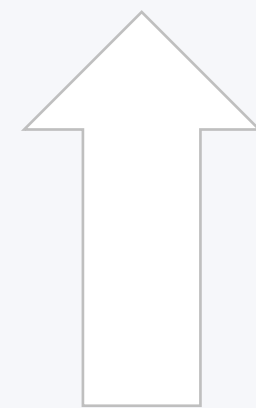
가장 긴 증가하는 부분 수열

45

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]
10	20	10	30	20	50



A[1]
10

A[1]	A[2]
10	20

A[1]	A[2]	A[3]
10	20	10

A[1]	A[2]	A[3]	A[4]
10	20	10	30

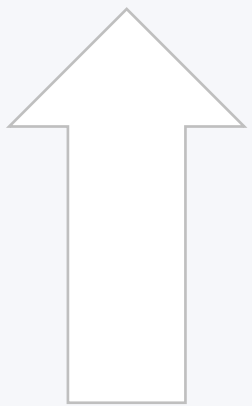
A[1]	A[2]	A[3]	A[4]	A[5]
10	20	10	30	20

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]
10	20	10	30	20	50



A[1]
10

$D[1] = 1$

A[1]	A[2]
10	20

$D[2] = 2$

A[1]	A[2]	A[3]
10	20	10

$D[3] = 1$

A[1]	A[2]	A[3]	A[4]
10	20	10	30

$D[4] = 3$

A[1]	A[2]	A[3]	A[4]	A[5]
10	20	10	30	20

$D[5] = 2$

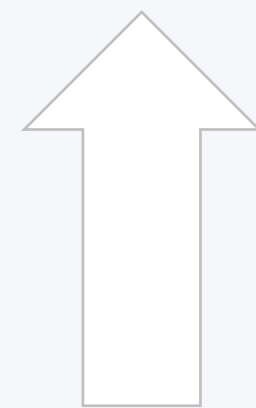
가장 긴 증가하는 부분 수열

47

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]
10	20	10	30	20	50



A[1]	D[1] = 1	A[6]	10 < 50
10		50	○

A[1]	A[2]	D[2] = 2	A[6]	20 < 50
10	20		50	○

A[1]	A[2]	A[3]	D[3] = 1	A[6]	10 < 50
10	20	10		50	○

A[1]	A[2]	A[3]	A[4]	D[4] = 3	A[6]	30 < 50
10	20	10	30		50	○

A[1]	A[2]	A[3]	A[4]	A[5]	D[5] = 2	A[6]	20 < 50
10	20	10	30	20		50	○

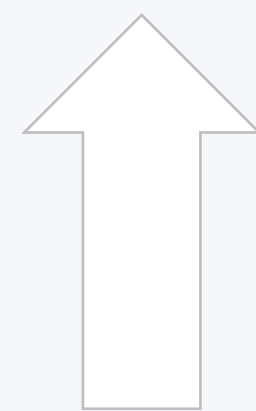
가장 긴 증가하는 부분 수열

48

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]
10	20	10	30	20	50



$D[6] = 4$

A[1]	A[6]
10	50

$D[1] = 1$ $10 < 50$ ○

A[1]	A[2]	A[6]
10	20	50

$D[2] = 2$ $20 < 50$ ○

A[1]	A[2]	A[3]	A[6]
10	20	10	50

$D[3] = 1$ $10 < 50$ ○

A[1]	A[2]	A[3]	A[4]	A[6]
10	20	10	30	50

$D[4] = 3$ $30 < 50$ ○

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]
10	20	10	30	20	50

$D[5] = 2$ $20 < 50$ ○

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

i	1	2	3	4	5	6
A[i]	10	20	10	30	20	50
D[i]	1					

가장 긴 증가하는 부분 수열

50

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

i	1	2	3	4	5	6
A[i]	10	20	10	30	20	50
D[i]	1	2				

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

i	1	2	3	4	5	6
A[i]	10	20	10	30	20	50
D[i]	1	2	1			

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

i	1	2	3	4	5	6
A[i]	10	20	10	30	20	50
D[i]	1	2	1	3		

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

i	1	2	3	4	5	6
A[i]	10	20	10	30	20	50
D[i]	1	2	1	3	2	

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

- $D[i] = A[1], \dots, A[i]$ 까지 수열이 있을 때, $A[i]$ 을 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

i	1	2	3	4	5	6
A[i]	10	20	10	30	20	50
D[i]	1	2	1	3	2	4

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

```
for (int i=0; i<n; i++) {  
    d[i] = 1;  
    for (int j=0; j<i; j++) {  
        if (a[j] < a[i] && d[i] < d[j]+1) {  
            d[i] = d[j]+1;  
        }  
    }  
}
```

가장 긴 증가하는 부분 수열

56

<https://www.acmicpc.net/problem/11053>

- 정답은 $D[1], \dots, D[N]$ 중의 최대값이 된다.

가장 긴 증가하는 부분 수열

<https://www.acmicpc.net/problem/11053>

- C: <https://gist.github.com/Baekjoon/1602b252bc8f1a1ee044>
- C++: <https://gist.github.com/Baekjoon/667ca791c2b9d5b1d2d2>
- Java: <https://gist.github.com/Baekjoon/9fc6b7ea58a9ea9968b2>

연속합

<https://www.acmicpc.net/problem/1912>

- n 개의 정수로 이루어진 임의의 수열이 주어진다.
- 우리는 이 중 연속된 몇 개의 숫자를 선택해서 구할 수 있는 합 중 가장 큰 합을 구하려고 한다.
- 단, 숫자는 한 개 이상 선택해야 한다.
- 예를 들어서 10, -4, 3, 1, 5, 6, -35, 12, 21, -1 이라는 수열이 주어졌다고 하자.
- 여기서 정답은 $12+21$ 인 33이 정답이 된다.

연속합

<https://www.acmicpc.net/problem/1912>

- $D[i]$ = i 번째 수로 끝나는 가장 큰 연속합
- 이렇게 식을 구했으면, i 번째 수에게 가능한 경우를 세야한다

연속합

<https://www.acmicpc.net/problem/1912>

- $D[i]$ = i 번째 수로 끝나는 가장 큰 연속합
- 이렇게 식을 구했으면, i 번째 수에게 가능한 경우를 세야한다
- i 번째 수에게 가능한 경우
 1. $i-1$ 번째 수의 연속합에 포함되는 경우
 2. 새로운 연속합을 시작하는 경우

연속합

<https://www.acmicpc.net/problem/1912>

- $D[i]$ = i 번째 수로 끝나는 가장 큰 연속합
- 이렇게 식을 구했으면, i 번째 수에게 가능한 경우를 세야한다
- i 번째 수에게 가능한 경우
 1. $i-1$ 번째 수의 연속합에 포함되는 경우
 - $D[i-1] + A[i]$
 2. 새로운 연속합을 시작하는 경우
 - $A[i]$
- 두 값 중에 어떤 값이 $D[i]$ 에 들어가야 할까? (최대값)
- $D[i] = \max(D[i-1] + A[i], A[i])$

연속합

<https://www.acmicpc.net/problem/1912>

- $D[i] = i$ 번째 수로 끝나는 가장 큰 연속합
- $D[i-1] + A[i] = 10 + -4 = 6$
- $A[i] = -4$

i	1	2	3	4	5	6	7	8	9	10
A[i]	10	-4	3	1	5	6	-35	12	21	-1
D[i]	10	6								

연속합

<https://www.acmicpc.net/problem/1912>

- $D[i]$ = i 번째 수로 끝나는 가장 큰 연속합
- $D[i-1] + A[i] = 6 + 3 = 9$
- $A[i] = 3$

i	1	2	3	4	5	6	7	8	9	10
A[i]	10	-4	3	1	5	6	-35	12	21	-1
D[i]	10	6	9							

연속합

65

<https://www.acmicpc.net/problem/1912>

- $D[i] = i$ 번째 수로 끝나는 가장 큰 연속합
- $D[i-1] + A[i] = 9 + 1 = 10$
- $A[i] = 1$

i	1	2	3	4	5	6	7	8	9	10
A[i]	10	-4	3	1	5	6	-35	12	21	-1
D[i]	10	6	9	10						

연속합

<https://www.acmicpc.net/problem/1912>

- $D[i] = i$ 번째 수로 끝나는 가장 큰 연속합
- $D[i-1] + A[i] = 10 + 5 = 15$
- $A[i] = 5$

i	1	2	3	4	5	6	7	8	9	10
A[i]	10	-4	3	1	5	6	-35	12	21	-1
D[i]	10	6	9	10	15					

연속합

<https://www.acmicpc.net/problem/1912>

- $D[i]$ = i 번째 수로 끝나는 가장 큰 연속합
- $D[i-1] + A[i] = 15 + 6 = 21$
- $A[i] = 6$

i	1	2	3	4	5	6	7	8	9	10
A[i]	10	-4	3	1	5	6	-35	12	21	-1
D[i]	10	6	9	10	15	21				

연속합

<https://www.acmicpc.net/problem/1912>

- $D[i] = i$ 번째 수로 끝나는 가장 큰 연속합
- $D[i-1] + A[i] = 21 + -35 = -14$
- $A[i] = -35$

i	1	2	3	4	5	6	7	8	9	10
A[i]	10	-4	3	1	5	6	-35	12	21	-1
D[i]	10	6	9	10	15	21	-14			

연속합

<https://www.acmicpc.net/problem/1912>

- $D[i] = i$ 번째 수로 끝나는 가장 큰 연속합
- $D[i-1] + A[i] = -14 + 12 = -2$
- $A[i] = 12$

i	1	2	3	4	5	6	7	8	9	10
A[i]	10	-4	3	1	5	6	-35	12	21	-1
D[i]	10	6	9	10	15	21	-14	12		

연속합

<https://www.acmicpc.net/problem/1912>

- $D[i]$ = i 번째 수로 끝나는 가장 큰 연속합
- $D[i-1] + A[i] = 12 + 21 = 33$
- $A[i] = 21$

i	1	2	3	4	5	6	7	8	9	10
A[i]	10	-4	3	1	5	6	-35	12	21	-1
D[i]	10	6	9	10	15	21	-14	12	33	

연속합

<https://www.acmicpc.net/problem/1912>

- $D[i] = i$ 번째 수로 끝나는 가장 큰 연속합
- $D[i-1] + A[i] = 33 + -1 = 32$
- $A[i] = -1$

i	1	2	3	4	5	6	7	8	9	10
A[i]	10	-4	3	1	5	6	-35	12	21	-1
D[i]	10	6	9	10	15	21	-14	12	33	32

- $D[i] = i$ 번째 수로 끝나는 가장 큰 연속합

[illegible]

연속합

<https://www.acmicpc.net/problem/1912>

- $D[i]$ = i 번째 수로 끝나는 가장 큰 연속합

i	1	2	3	4	5	6	7	8	9	10
A[i]	10	-4	-10	7	-35	12	21	-10	5	8
D[i]	10	6	-4	7	-28	12	34	24	29	37

연속합

<https://www.acmicpc.net/problem/1912>

```
for (int i=0; i<n; i++) {  
    d[i] = a[i];  
    if (i == 0) continue;  
    if (d[i] < d[i-1] + a[i]) {  
        d[i] = d[i-1] + a[i];  
    }  
}
```

연속합

<https://www.acmicpc.net/problem/1912>

- C: <https://gist.github.com/Baekjoon/6577cbbc4ebeda00604c>
- C++: <https://gist.github.com/Baekjoon/dc368ddfc7c138a6411f>
- Java: <https://gist.github.com/Baekjoon/885882aa75ec74535c45>

합분해

76

<https://www.acmicpc.net/problem/2225>

- 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수

합분해

<https://www.acmicpc.net/problem/2225>

- 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N]$ = 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $? + ? + ? + ? + \dots + ? + L = N$
- 위의 식이 나타내는 값: $D[K][N]$
- $? + ? + ? + ? + \dots + ? = N - L$
- 위의 식이 나타내는 값: $D[K-1][N-L]$
- $D[K][N] = \sum D[K-1][N-L] \ (0 \leq L \leq N)$

합분해

<https://www.acmicpc.net/problem/2225>

```
d[0][0] = 1LL;
for (int i=1; i<=k; i++) {
    for (int j=0; j<=n; j++) {
        for (int l=0; l<=j; l++) {
            d[i][j] += d[i-1][j-l];
            d[i][j] %= mod;
        }
    }
}
```

합분해

<https://www.acmicpc.net/problem/2225>

- C/C++: <https://gist.github.com/Baekjoon/a334580d1729037f5fb1>
- Java: <https://gist.github.com/Baekjoon/354ed0a3657ecbf00c67>

합분해

<https://www.acmicpc.net/problem/2225>

- 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N]$ = 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N] = \sum D[K-1][N-L] \ (0 \leq L \leq N)$
- $D[K][N] = \sum D[K-1][N-L] \ (0 \leq N-L \leq N)$
- $D[K][N] = \sum D[K-1][L] \ (0 \leq L \leq N)$

합분해

<https://www.acmicpc.net/problem/2225>

- 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N]$ = 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N] = \sum D[K-1][L] \ (0 \leq L \leq N)$
- $D[K][N] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1] + D[K-1][N]$
- $D[K][N-1] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1]$

합분해

<https://www.acmicpc.net/problem/2225>

- 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N]$ = 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N] = \sum D[K-1][L] \ (0 \leq L \leq N)$
- $D[K][N] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1] + D[K-1][N]$
- $D[K][N-1] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1]$
- $D[K][N] = D[K][N-1] + D[K-1][N]$

합분해

<https://www.acmicpc.net/problem/2225>

- C++: <https://gist.github.com/Baekjoon/307eeb5e498f68a5f0fab64f6ae0bd05>
- Java: <https://gist.github.com/Baekjoon/3b53e9288ac9d652d8a6e34688c272c8>