

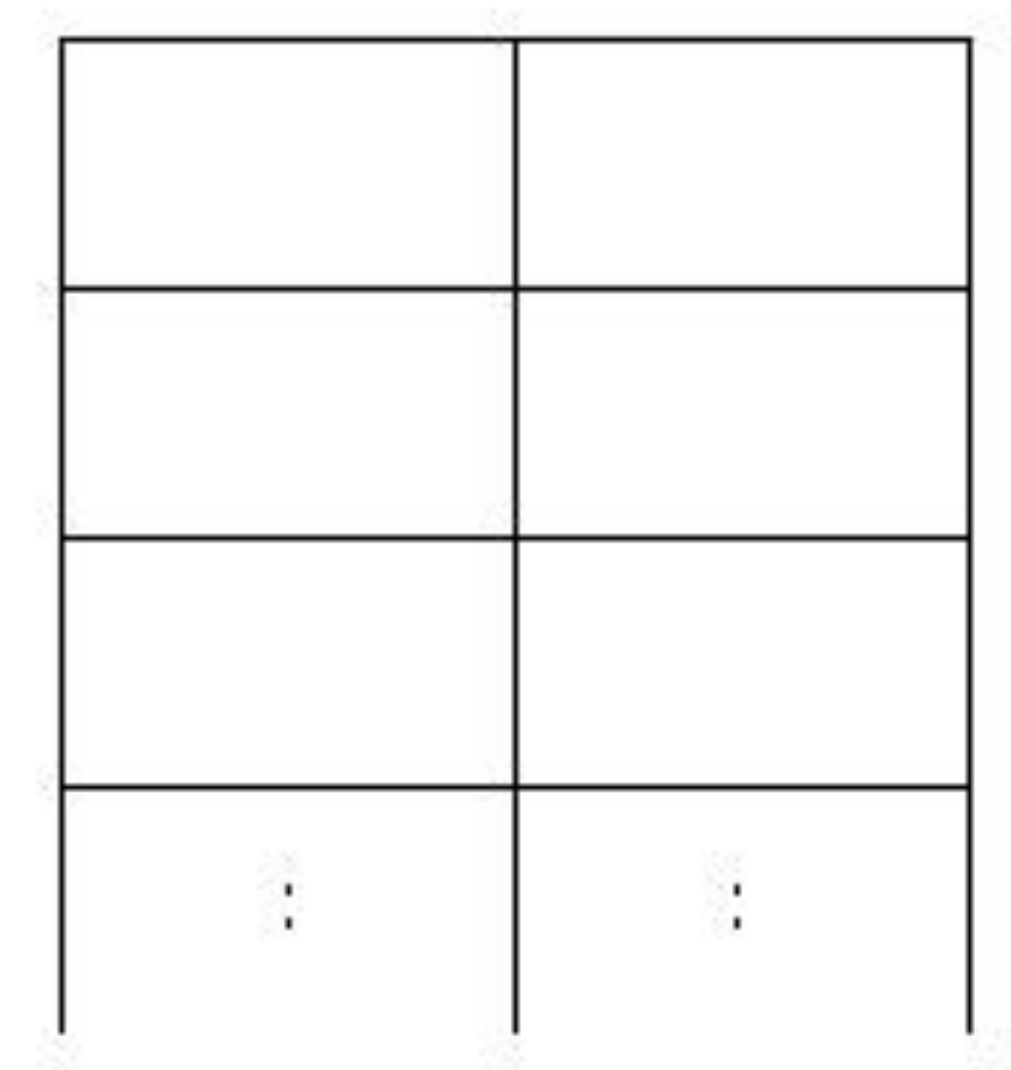
다이나믹 프로그래밍 1 (도전)

최백준 choi@startlink.io

동물원

<https://www.acmicpc.net/problem/1309>

- 가로로 두 칸, 세로로 N 칸인 동물원이 있다
- 가로, 세로로 붙어 있게 배치하면 안된다
- 가능한 배치의 수



동물원

<https://www.acmicpc.net/problem/1309>

- $D[i]$ = 세로 크기가 i 인 동물원을 채우는 방법의 수, 단 i 번째 줄에는 동물이 있어야 한다.
- i 번째 줄의 이전에 동물이 있는 줄은 어디일까?

1		
2		
...		
$i-3$		
$i-2$		
$i-1$		
i	동	물

동물원

<https://www.acmicpc.net/problem/1309>

- $D[i]$ = 세로 크기가 i 인 동물원을 채우는 방법의 수, 단 i 번째 줄에는 동물이 있어야 한다.
- i 번째 줄의 이전에 동물이 있는 줄은 어디일까?
- $i-1$ 번째 줄일 수도 있고, $i-2$ 일 수도 있고, \dots , 2 , 1 일 수도 있다.

1	동물	동물
2	동물	동물
...		
$i-3$		
$i-2$	동물	동물
$i-1$	동물	동물
i	동물	동물

동물원

<https://www.acmicpc.net/problem/1309>

- $D[i]$ = 세로 크기가 i 인 동물원을 채우는 방법의 수, 단 i 번째 줄에는 동물이 있어야 한다.
- i 번째 줄의 이전에 동물이 있는 줄은 어디일까?
- $i-1$ 번째 줄일 수도 있고, $i-2$ 일 수도 있고, \dots , 2 , 1 일 수도 있다.
- $D[i-1] + D[i-2] + \dots + D[2] + D[1]$

1	동물	동물
2	동물	동물
...		
$i-3$		
$i-2$	동물	동물
$i-1$	동물	동물
i	동물	동물

동물원

<https://www.acmicpc.net/problem/1309>

- $D[i]$ = 세로 크기가 i 인 동물원을 채우는 방법의 수, 단 i 번째 줄에는 동물이 있어야 한다.
- i 번째 줄의 이전에 동물이 있는 줄은 어디일까?
- $i-1$ 번째 줄일 수도 있고, $i-2$ 일 수도 있고, \dots , 2 , 1 일 수도 있다.
- $D[i-1] + D[i-2] + \dots + D[2] + D[1]$
- 이제 붙어 있게 배치할 수 없는 문제의 조건을 추가해보자

1	동물	동물
2	동물	동물
...		
$i-3$		
$i-2$	동물	동물
$i-1$	동물	동물
i	동물	동물

동물원

<https://www.acmicpc.net/problem/1309>

- $D[i]$ = 세로 크기가 i 인 동물원을 채우는 방법의 수, 단 i 번째 줄에는 동물이 있어야 한다.
- $i-1$ 번째 줄은 한 가지
- 1번째부터 $i-2$ 번째 줄은
- 두 가지씩 가능하다.

1	O, X	X, O
2	O, X	X, O
...		
$i-3$		
$i-2$	O, X	X, O
$i-1$	X	O
i	O	X

1	O, X	X, O
2	O, X	O, X
...		
$i-3$		
$i-2$	O, X	O, X
$i-1$	O	X
i	X	O

동물원

<https://www.acmicpc.net/problem/1309>

- $D[i]$ = 세로 크기가 i 인 동물원을 채우는 방법의 수, 단 i 번째 줄에는 동물이 있어야 한다.
- i 번째 줄의 이전에 동물이 있는 줄은 어디일까?
- $i-1$ 번째 줄일 수도 있고, $i-2$ 일 수도 있고, \dots , 2, 1일 수도 있다.
- $D[i-1] + 2 \times D[i-2] + \dots + 2 \times D[2] + 2 \times D[1]$
- $D[i-1] + 2(D[i-2] + \dots + D[2] + D[1])$
- 이 방법을 이용해서 구현하면 $O(N^2)$ 에 해결할 수 있다.
- 하지만 이 문제의 제한은 $N \leq 100,000$ 이다.

1	동물	동물
2	동물	동물
...		
i-3		
i-2	동물	동물
i-1	동물	동물
i	동물	동물

동물원

<https://www.acmicpc.net/problem/1309>

- $D[i]$ = 세로 크기가 i 인 동물원을 채우는 방법의 수, 단 i 번째 줄에는 동물이 있어야 한다.
- $S[i] = D[0] + D[1] + D[2] + \dots + D[i]$ 를 저장한다면
- $D[i] = D[i-1] + 2(D[i-2] + \dots + D[2] + D[1])$ 를
- $D[i] = D[i-1] + 2 \times S[i-2]$ 로 구현할 수 있다.
- 이를 이용하면 $O(N)$ 이 가능하다.

1	동물	동물
2	동물	동물
...		
$i-3$		
$i-2$	동물	동물
$i-1$	동물	동물
i	동물	동물

동물원

10

<https://www.acmicpc.net/problem/1309>

- 소스: <http://codeplus.codes/aa648f9f4b55408fb022225cb79e50fd>

RGB 거리 2

<https://www.acmicpc.net/problem/17404>

- RGB거리에 사는 사람들은 집을 빨강, 초록, 파랑중에 하나로 칠하려고 한다
- 또한, 그들은 모든 이웃은 같은 색으로 칠할 수 없다는 규칙도 정했다
- 집 i 의 이웃은 집 $i-1$ 과 집 $i+1$ 이고, 첫 집과 마지막 집도 이웃이다.
- 처음 집과 마지막 집은 이웃이 아니다
- 각 집을 빨강으로 칠할 때 드는 비용, 초록으로 칠할 때 드는 비용, 파랑으로 드는 비용이 주어질 때, 모든 집을 칠하는 비용의 최솟값을 구하는 문제

RGB 거리 2

<https://www.acmicpc.net/problem/17404>

- $D[i][j]$ = i 번 집을 색 j 로 칠했을 때, $1 \sim i$ 번 집을 칠하는 비용의 최소값
 - $j = 0 \rightarrow$ 빨강
 - $j = 1 \rightarrow$ 초록
 - $j = 2 \rightarrow$ 파랑
- $D[i][j]$ = i 번 집을 색 j 로 칠했을 때, $1 \sim i$ 번 집을 칠하는 비용의 최소값

RGB 거리 2

<https://www.acmicpc.net/problem/17404>

- $D[i][0] = \min(D[i-1][1], D[i-1][2]) + A[i][0]$
- $D[i][1] = \min(D[i-1][0], D[i-1][2]) + A[i][1]$
- $D[i][2] = \min(D[i-1][0], D[i-1][1]) + A[i][2]$

RGB 거리 2

<https://www.acmicpc.net/problem/17404>

- 하지만, 이런 식으로는 정답을 구할 수 없다.
- 1번 집과 N번 집이 같은 색이 되지 않아야 한다.
- 1번 집과 N번 집의 색은 다음 6가지가 가능하다.
 - 빨강, 초록
 - 빨강, 파랑
 - 초록, 빨강
 - 초록, 파랑
 - 파랑, 빨강
 - 파랑, 초록

RGB 거리 2

<https://www.acmicpc.net/problem/17404>

- 1번 집의 색을 빨강으로 고정하고 답을 구한다면, 다음 6가지 중 2가지를 구할 수 있다.
- 1번 집과 N번 집의 색은 다음 6가지가 가능하다.
 - 빨강, 초록
 - 빨강, 파랑
 - 초록, 빨강
 - 초록, 파랑
 - 파랑, 빨강
 - 파랑, 초록

RGB 거리 2

<https://www.acmicpc.net/problem/17404>

- 1번 집의 색을 초록으로 고정하고 답을 구한다면, 다음 6가지 중 2가지를 구할 수 있다.
- 1번 집과 N번 집의 색은 다음 6가지가 가능하다.
 - 빨강, 초록
 - 빨강, 파랑
 - 초록, 빨강
 - 초록, 파랑
 - 파랑, 빨강
 - 파랑, 초록

RGB 거리 2

<https://www.acmicpc.net/problem/17404>

- 1번 집의 색을 파랑으로 고정하고 답을 구한다면, 다음 6가지 중 2가지를 구할 수 있다.
- 1번 집과 N번 집의 색은 다음 6가지가 가능하다.
 - 빨강, 초록
 - 빨강, 파랑
 - 초록, 빨강
 - 초록, 파랑
 - 파랑, 빨강
 - 파랑, 초록

RGB 거리 2

<https://www.acmicpc.net/problem/17404>

- 1번 집의 색상을 미리 정해놓은 다음, 다이나믹을 3번 수행해서 정답을 구할 수 있다.

RGB 거리 2

<https://www.acmicpc.net/problem/17404>

- 소스: <http://codeplus.codes/5f0c113a1e554c15a59d2cd6bd34d22c>

합분해

20

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

- 0부터 N 까지의 정수 K 개를 더해서 그 합이 N 이 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
- $D[K][N] = \sum D[K-1][N-L] \ (0 \leq L \leq N)$
- $D[K][N] = \sum D[K-1][N-L] \ (0 \leq N-L \leq N)$
- $D[K][N] = \sum D[K-1][L] \ (0 \leq L \leq N)$

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
- $D[K][N] = \sum D[K-1][L] \ (0 \leq L \leq N)$
- $D[K][N] = \mathbf{D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1]} + D[K-1][N]$
- $D[K][N-1] = \mathbf{D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1]}$
- $D[K][N] = D[K][N-1] + D[K-1][N]$

합분해

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

- $D[K][N] = \sum D[K-1][L] \ (0 \leq L \leq N)$
- $D[K][N] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1] + D[K-1][N]$
- $D[K][N-1] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1]$

				N-1	N		
K-1							
K							

합분해

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

- $D[K][N] = \sum D[K-1][L] \ (0 \leq L \leq N)$
- $D[K][N] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1] + D[K-1][N]$
- $D[K][N-1] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1]$

				N-1	N		
K-1							
K							

합분해

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

- $D[K][N] = \sum D[K-1][L] \ (0 \leq L \leq N)$
- $D[K][N] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1] + D[K-1][N]$
- $D[K][N-1] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1]$

				N-1	N		
K-1							
K							

합분해

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

- $D[K][N] = \sum D[K-1][L] \ (0 \leq L \leq N)$
- $D[K][N] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1] + D[K-1][N]$
- $D[K][N-1] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1]$
- $D[K][N] = D[K][N-1] + D[K-1][N]$

				N-1	N		
K-1							
K							

합분해

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

- 소스: <http://codeplus.codes/4a6b3ffc40154b3e8dd4aa0dc8870855>

합분해

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

- $D[K][N] = D[K][N-1] + D[K-1][N]$ 의 경우
- 일차원 다이나믹으로 바꿀 수 있다.
- $D2[N] = D[K][N]$ 을 넣을 예정

			N-3	N-2	N-1	N	
K-1							
K							

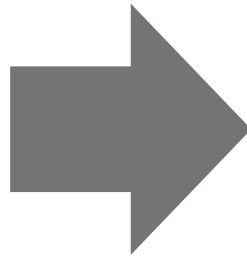
			N-3	N-2	N-1	N	
K-1							

합분해

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

- $D[K][N] = D[K][N-1] + D[K-1][N]$ 의 경우
- $D2[N-2] = D2[N-3] + D2[N-2]$

			N-3	N-2	N-1	N	
D2							



			N-3	N-2	N-1	N	
D2							

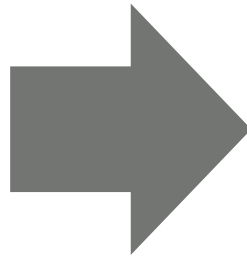
			N-3	N-2	N-1	N	
K-1							
K							

합분해

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

- $D[K][N] = D[K][N-1] + D[K-1][N]$ 의 경우
- $D2[N-1] = D2[N-2] + D2[N-1]$

			N-3	N-2	N-1	N	
D2							



			N-3	N-2	N-1	N	
D2							

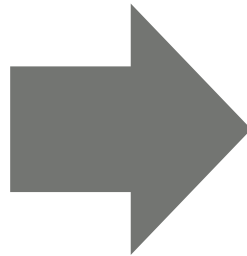
			N-3	N-2	N-1	N	
K-1							
K							

합분해

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

- $D[K][N] = D[K][N-1] + D[K-1][N]$ 의 경우
- $D2[N] = D2[N-1] + D2[N]$

			N-3	N-2	N-1	N	
D2							



			N-3	N-2	N-1	N	
D2							

			N-3	N-2	N-1	N	
K-1							
K							

합분해

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

- 소스: <http://codeplus.codes/7becba12ef1b42079b1163544b5be6b4>

끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.