DP 심화: 비트 DP & 덱 DP
(Bit DP & Deque DP)

신촌 연합 겨울캠프 중급 9회차

UNIST 한동규 (queued_q)

비트 DP

(Bit DP)

비트 연산자

- i번째 비트 구하기
- i번째 비트를 0/1로 설정하기
- 마지막 n비트 구하기
- i번째 비트 반전

- 연산자 우선순위 주의!

~ !

\+ −

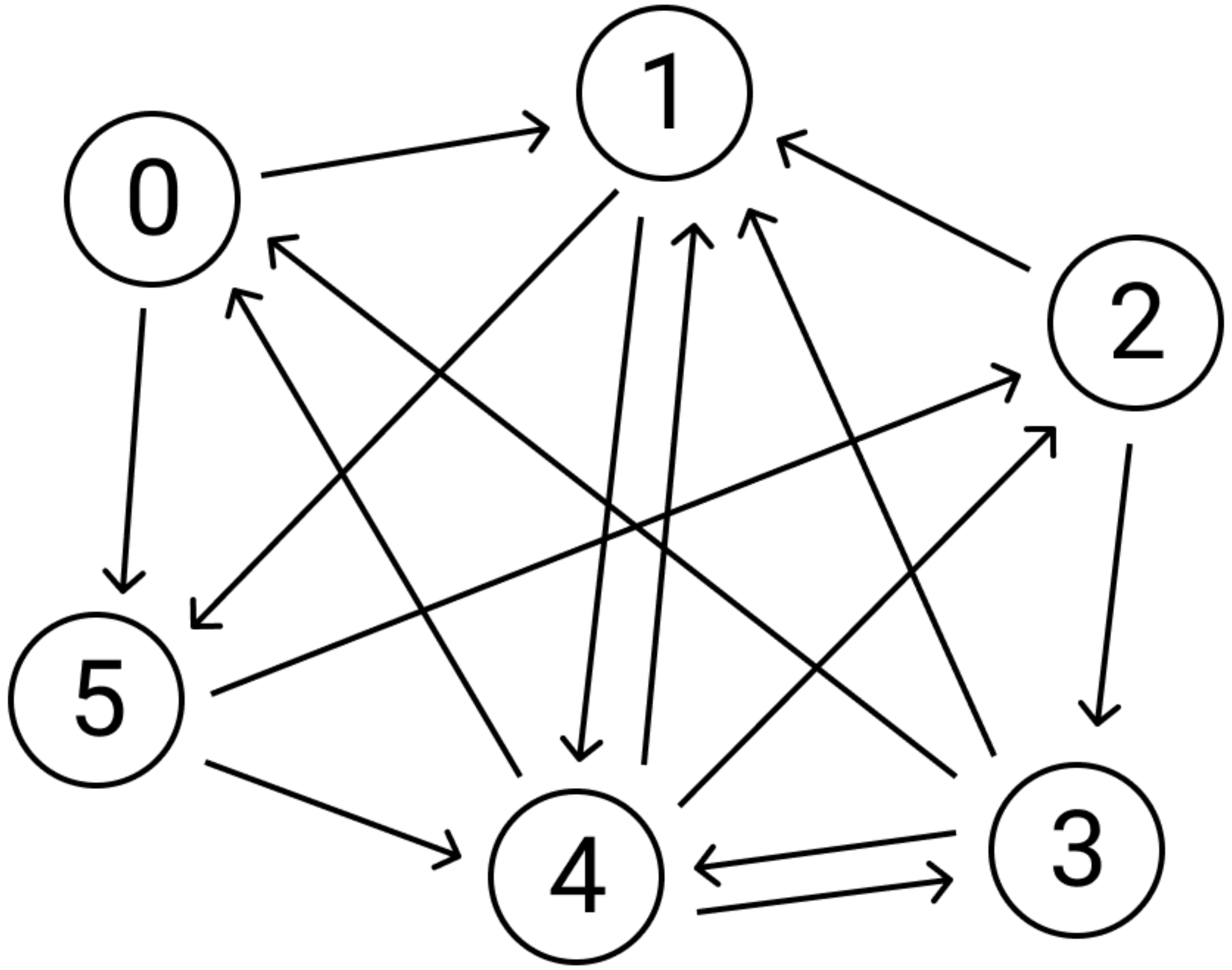
<< >>

== !=

& ^ |

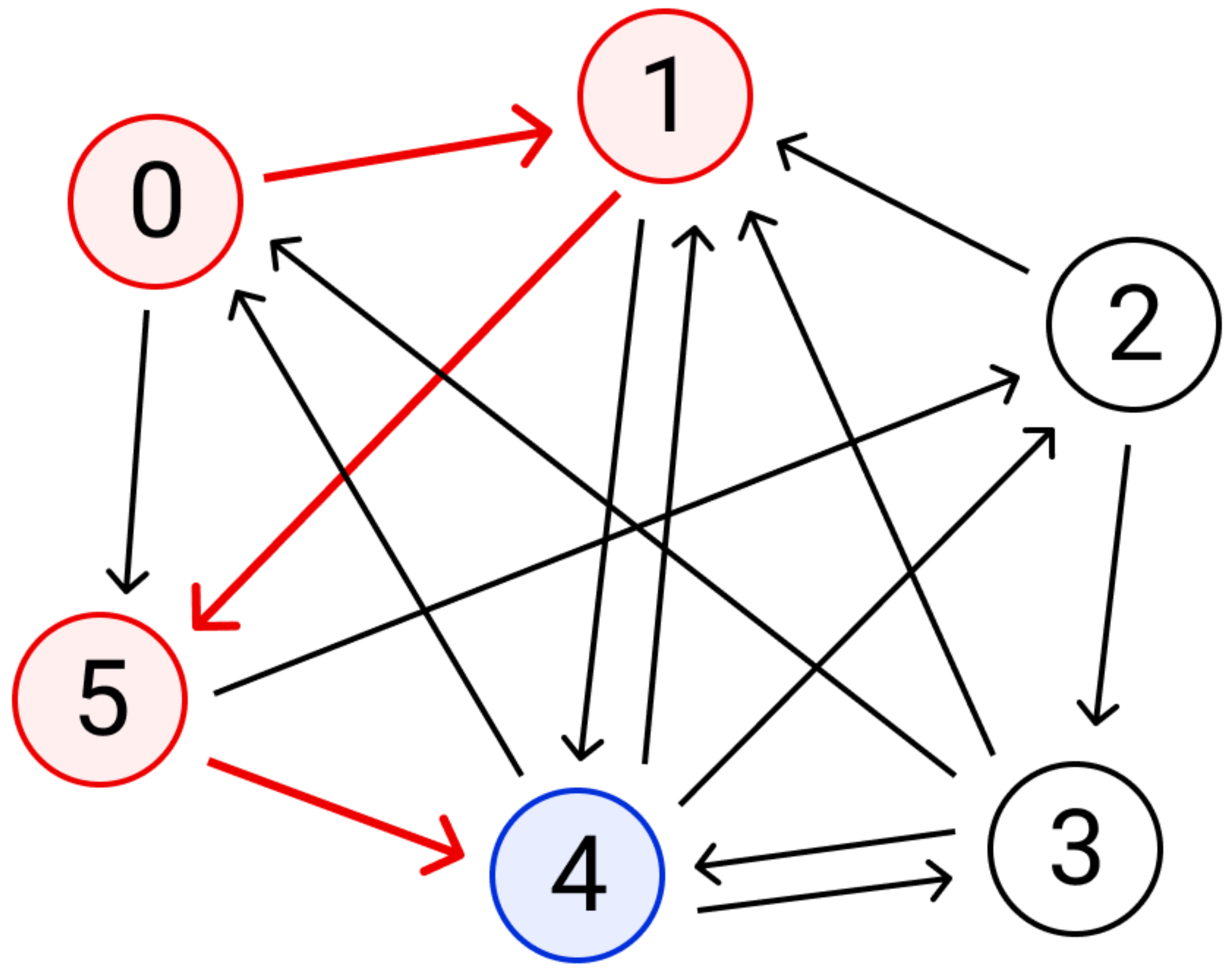
연습문제) BOJ 2098 외판원 순회

- 도시 N ≤ 16 개와 도시를 잇는 길이 가중치 있는 방향그래프 형태로 주어진다.
- 외판원이 모든 도시를 한 번씩 방문하고 다시 원래 도시로 돌아오려고 한다.
- 최소 비용의 순회 경로는?



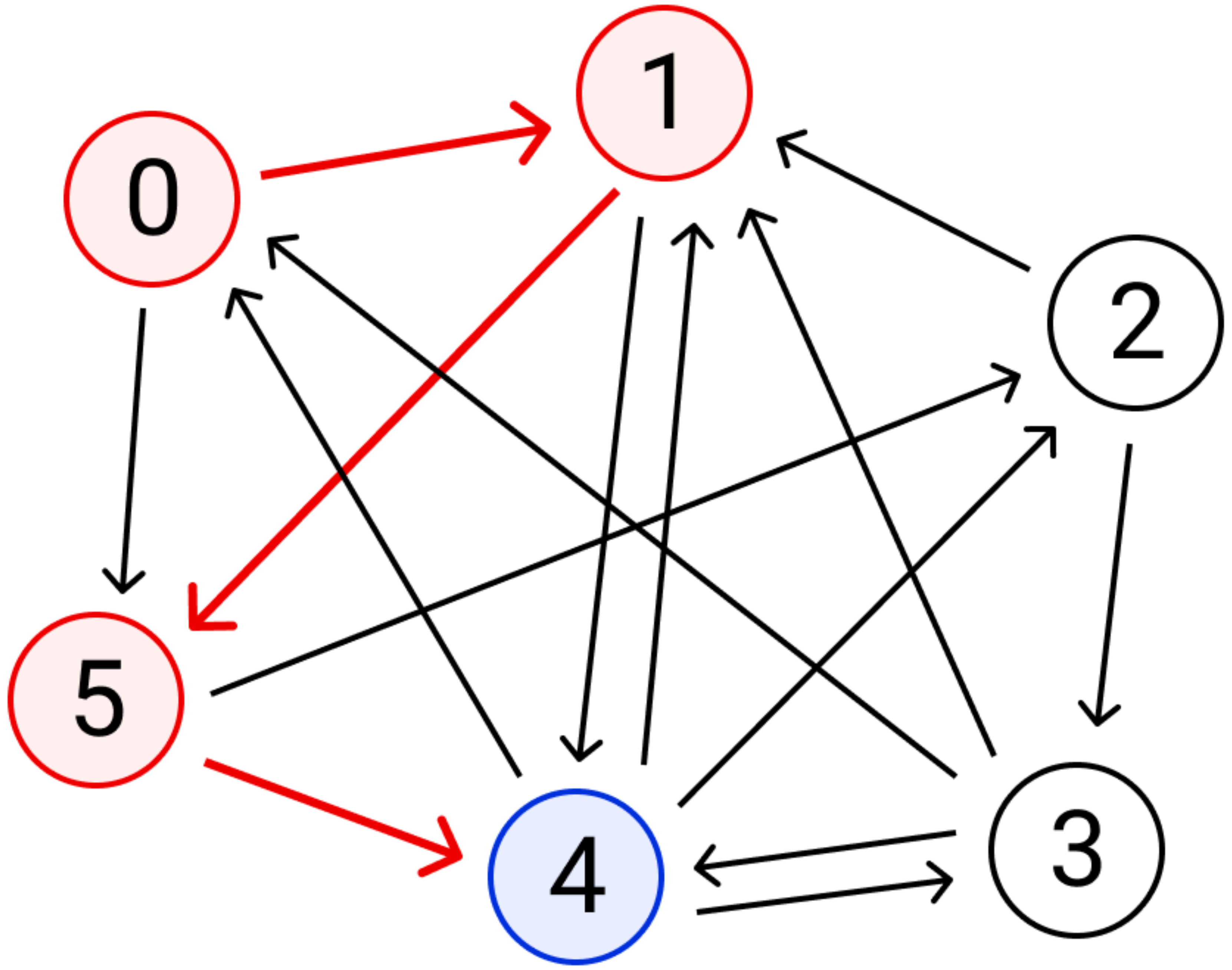
연습문제) BOJ 2098 외판원 순회

- 나이브한 접근: 완전 탐색 (백트래킹)
- 경우의 수: O(N!)
- 16! = 20,922,789,888,000



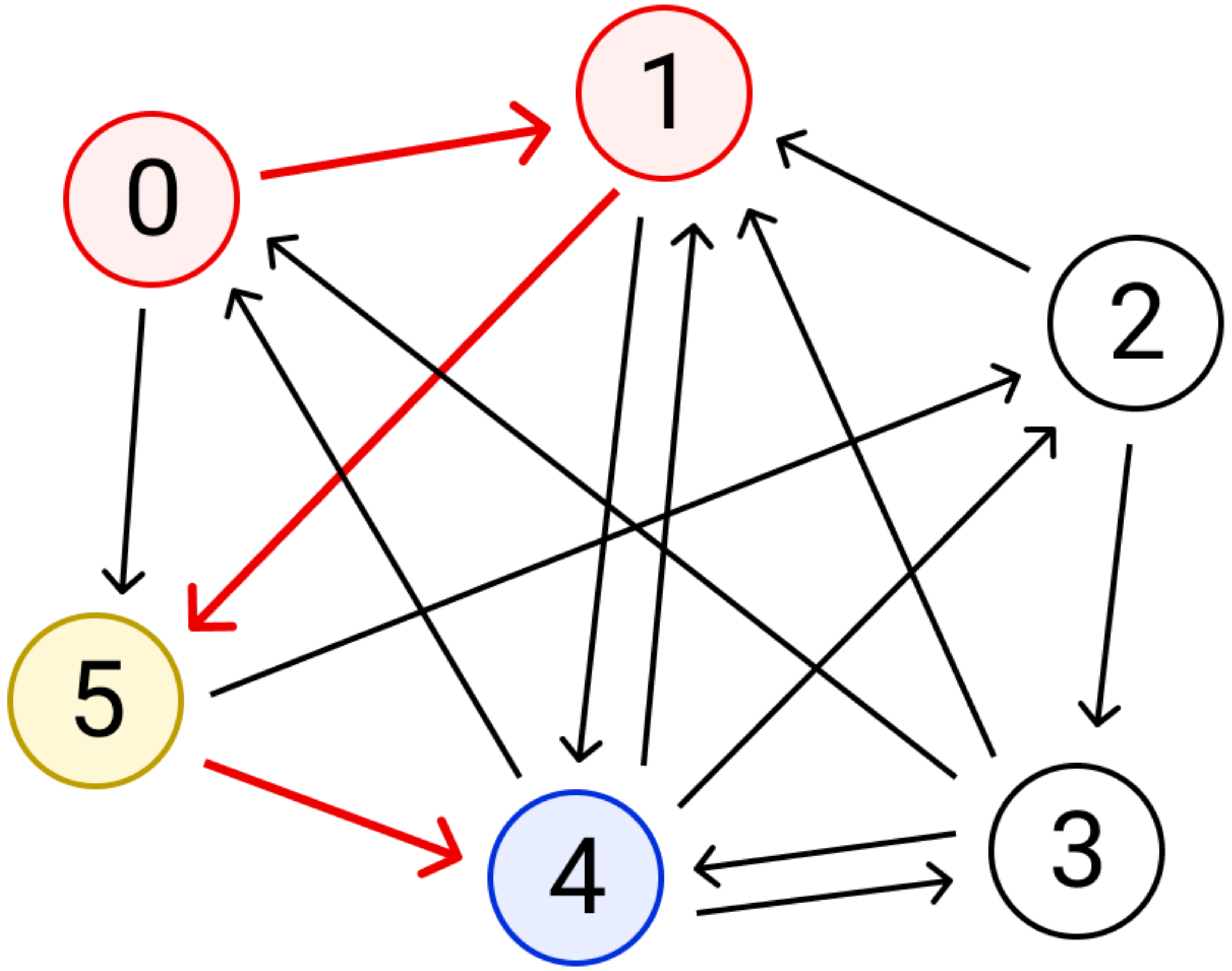
연습문제) BOJ 2098 외판원 순회

- 관찰: 백트래킹을 진행하는 중간 과정을 보면, 앞으로의 경로를 설정하는 데 불필요한 정보가 있다!
- 원래 상태: (현재까지 지나온 도시의 경로)
- 새 상태: (현재까지 지나온 도시의 집합, 현재 도시)
- 경우의 수: $O(2^N \cdot N)$
- $2^{16} \cdot 16 = 1{,}048{,}576$



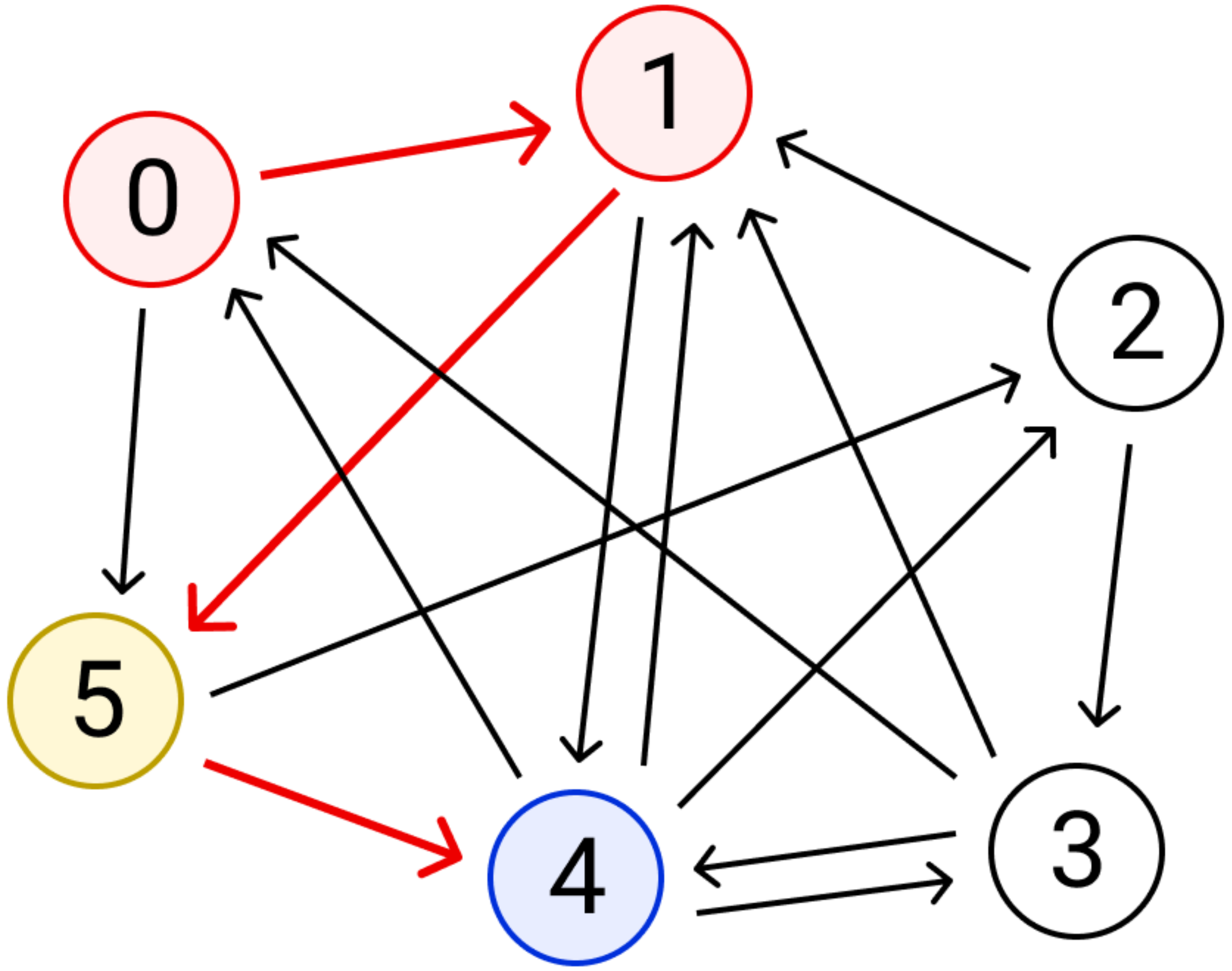
연습문제) BOJ 2098 외판원 순회

- D[vst][i] : vst 집합의 도시를 모두 방문하고 i번 도시에 도착하는 경로의 최소 비용
- D[vst][i] = min_{W[j][i] ≠ 0} D[vst - (1<<i)][j] + W[j][i]
- 단, vst의 i, j번째 비트는 1이어야 함



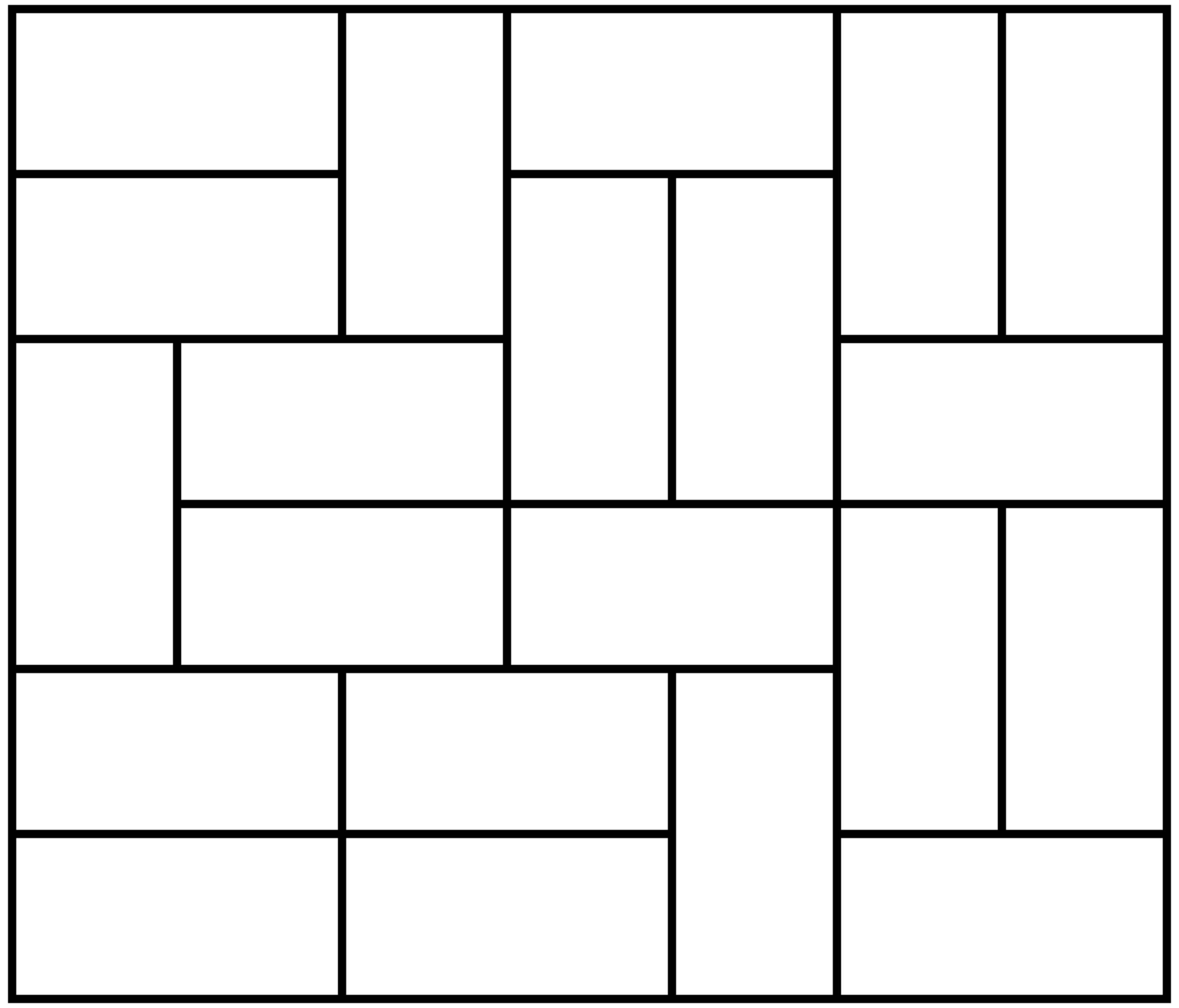
연습문제) BOJ 2098 외판원 순회

- `D[vst][i]` : `vst` 집합의 도시를 모두 방문하고 `i`번 도시에 도착하는 경로의 최소 비용
- `D[vst][i] = min_{W[j][i] ≠ 0} D[vst - (1<<i)][j] + W[j][i]`
- 단, `vst`의 `i`, `j`번째 비트는 1이어야 함
- `vst`를 `0`부터 1<<N 직전까지 그냥 증가시켜도 된다.
- 순서를 고려하기 번거로우면 top-down DP를 짜도 된다.
- 시간복잡도: $O(2^N \cdot N^2)$



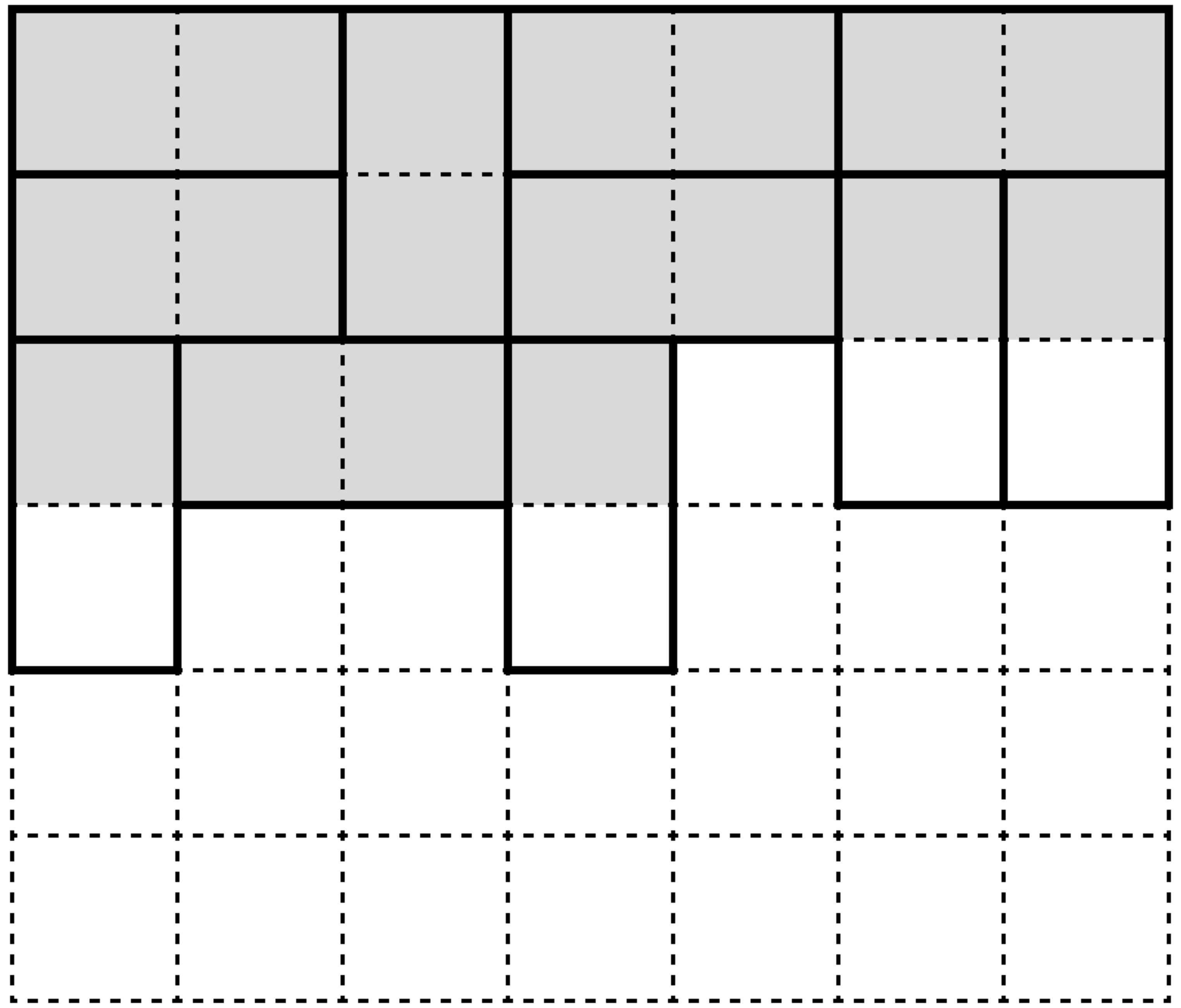
연습문제) BOJ 1648 격자판 채우기

- 격자판의 세로, 가로 크기 N, M ≤ 14 가 주어진다.
- 격자판을 2×1 크기 도미노로 채우는 경우의 수는?



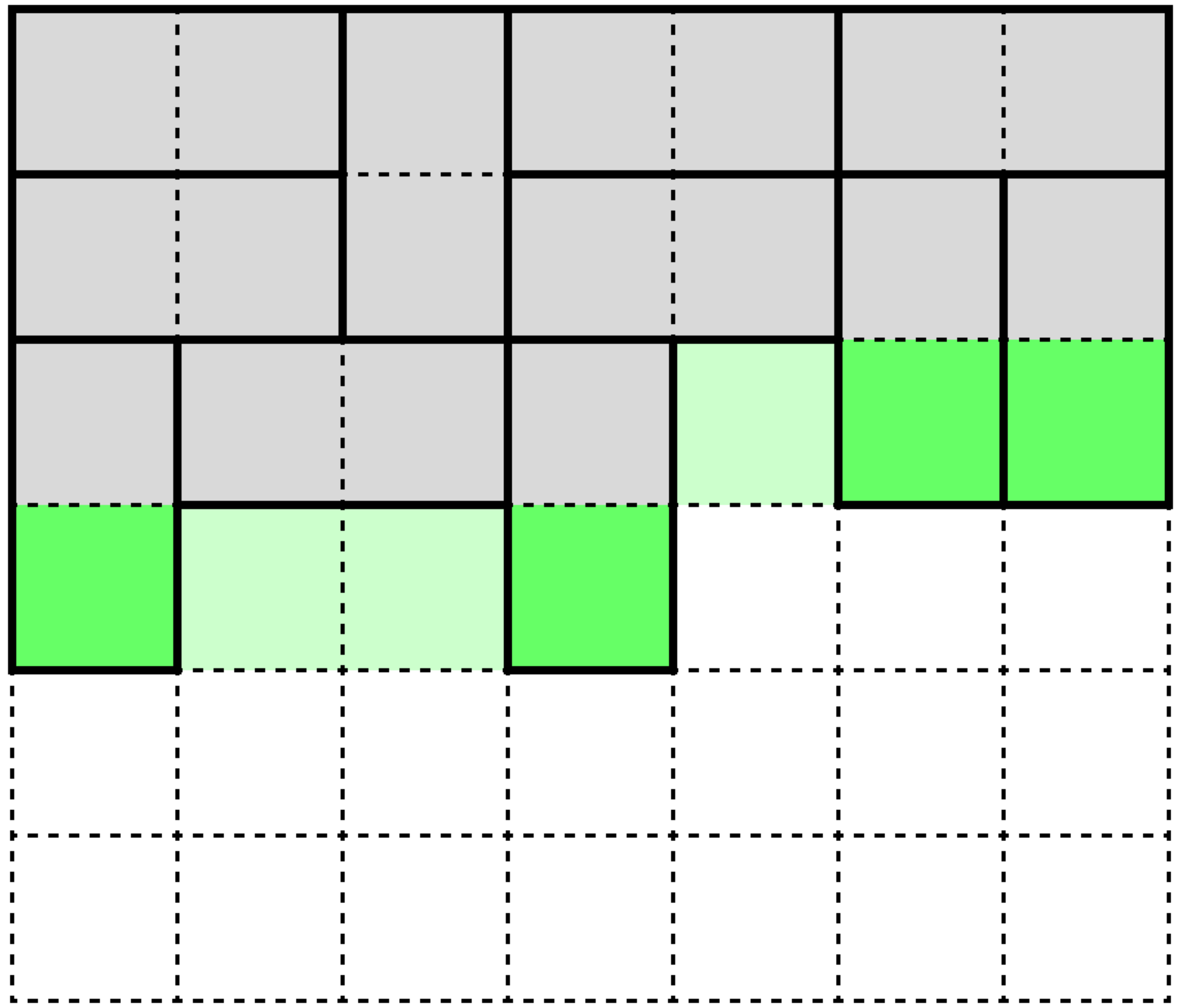
연습문제) BOJ 1648 격자판 채우기

- 백트래킹 풀이: 가장 윗줄, 가장 왼쪽에 있는 빈칸에 도미노를 원하는 방향으로 놓기
- 상태: 도미노를 놓은 방향 (수직, 수평)의 배열
- 경우의 수: $O(2^{NM/2})$



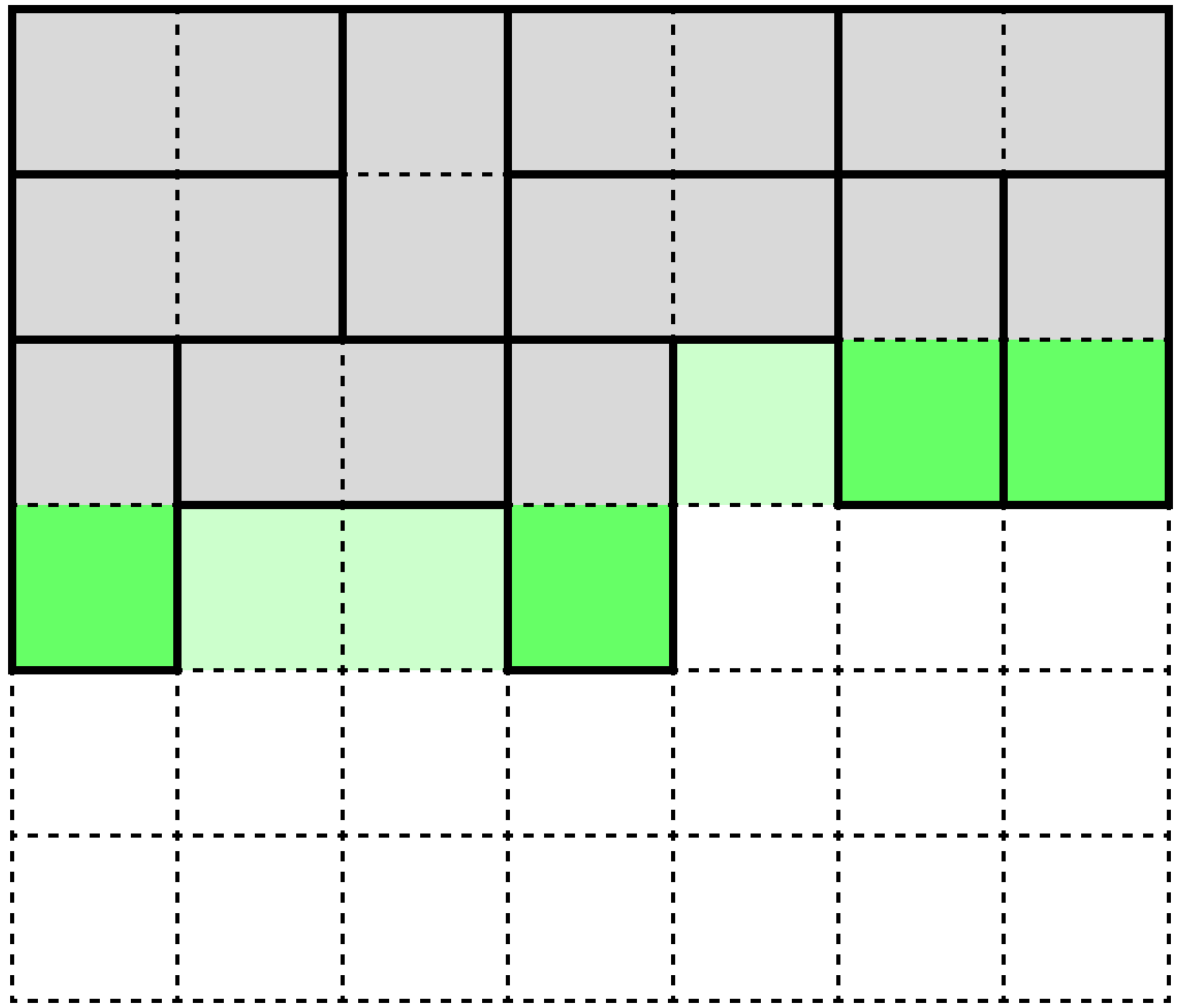
연습문제) BOJ 1648 격자판 채우기

- 관찰: 다음에 도미노를 채우려는 위치가 i번 칸일 때,
  - [0, i) 번 칸은 전부 채워져 있다.
  - [i, i+M) 번 칸은 채워져 있을 수도, 비어 있을 수도 있다.
  - [i+M, NM) 번 칸은 전부 비어 있다.
- 상태: [i, i+M)번 칸이 채워진 패턴
- 경우의 수: $O(NM \cdot 2^M)$



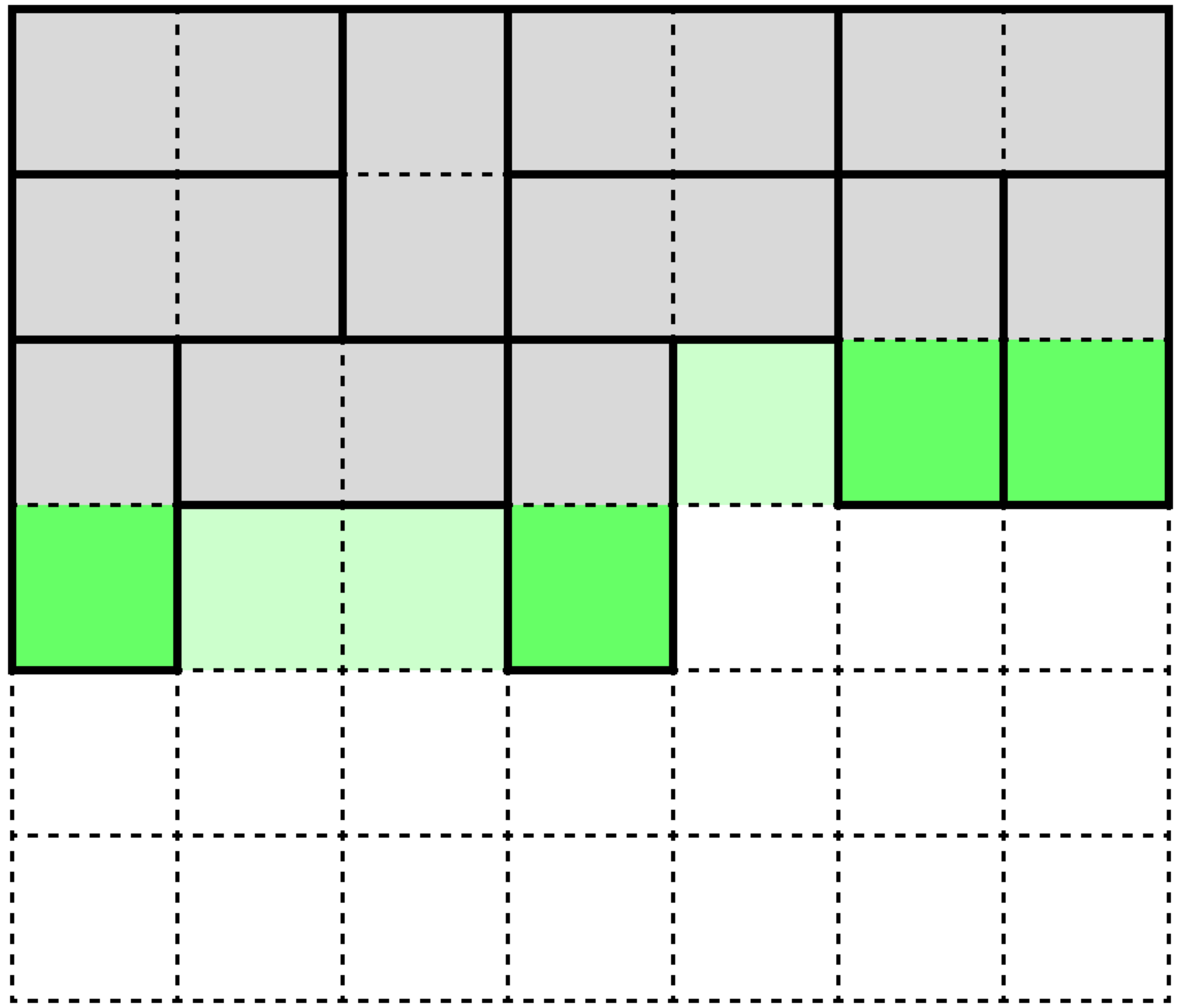
연습문제) BOJ 1648 격자판 채우기

- DP 식을 어떻게?
- 현재 상태로 올 수 있는 이전 상태를 알 때: 현재까지의 경우의 수로 정의
- 현재 상태에서 갈 수 있는 다음 상태를 알 때: 앞으로의 경우의 수로 정의
- DP 테이블을 역순으로 채우거나, top-down DP



연습문제) BOJ 1648 격자판 채우기

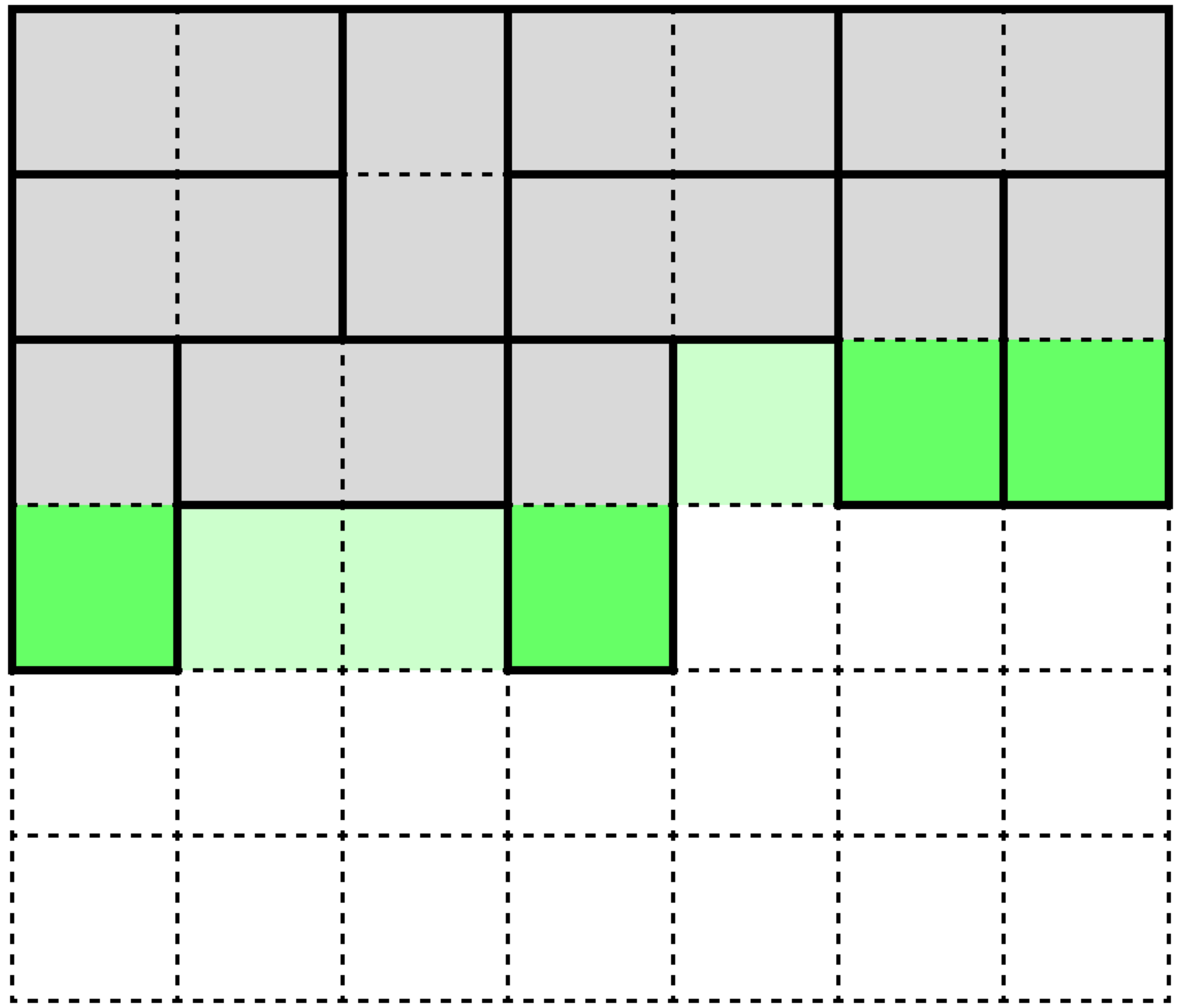
- `D[i][state]` : i번 이전 칸까지 도미노를 놓았고, [i, i+M) 번 칸이 state의 패턴으로 채워져 있을 때, 남은 칸을 채우는 경우의 수



연습문제) BOJ 1648 격자판 채우기

- `D[i][state]` : i번 이전 칸까지 도미노를 놓았고, [i, i+M) 번 칸이 state의 패턴으로 채워져 있을 때, 남은 칸을 채우는 경우의 수

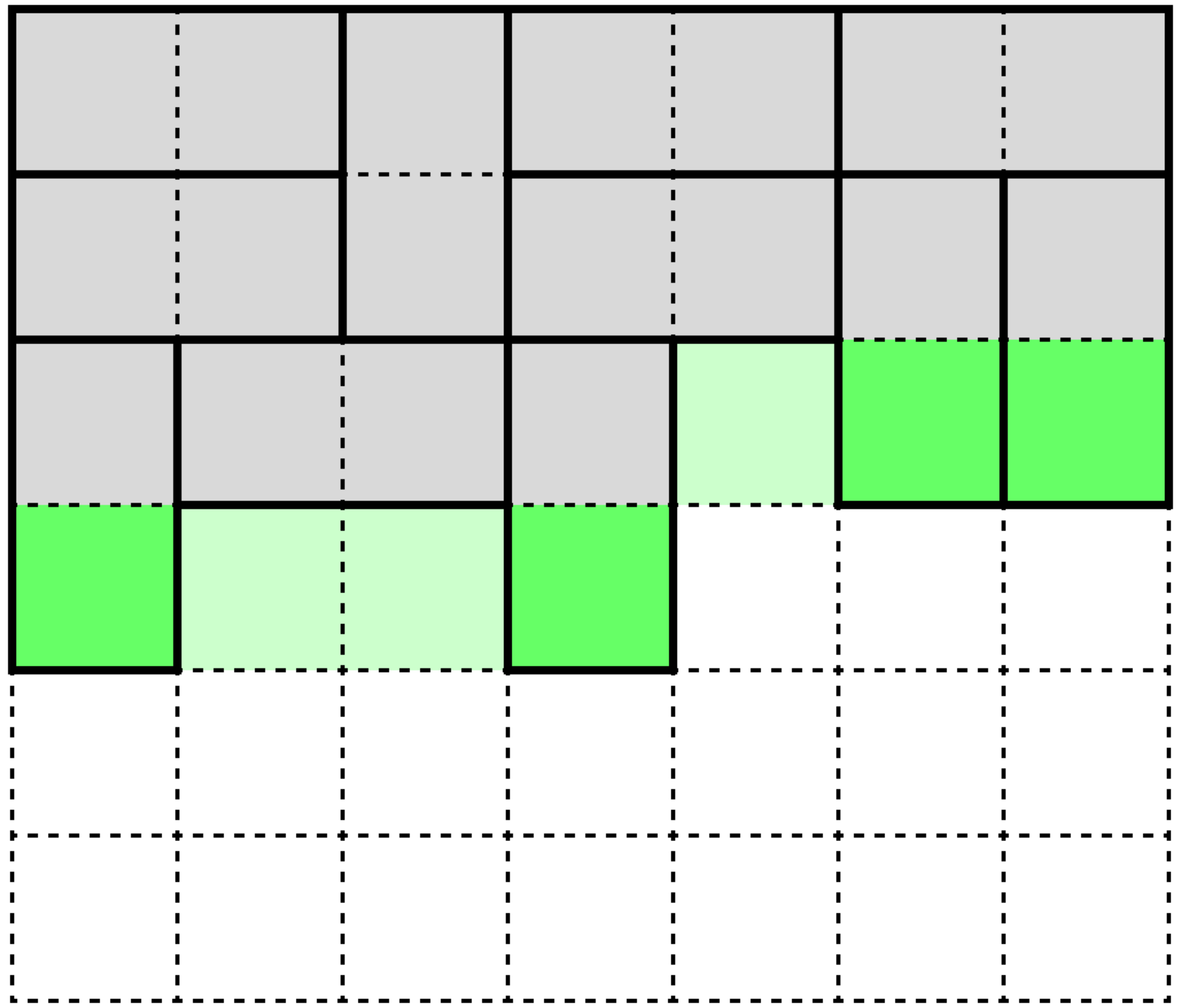
1. i번 칸이 채워져 있는 경우
   - `D[i][state] = D[i+1][state>>1]`



연습문제) BOJ 1648 격자판 채우기

- `D[i][state]` : i번 이전 칸까지 도미노를 놓았고, [i, i+M) 번 칸이 state의 패턴으로 채워져 있을 때, 남은 칸을 채우는 경우의 수

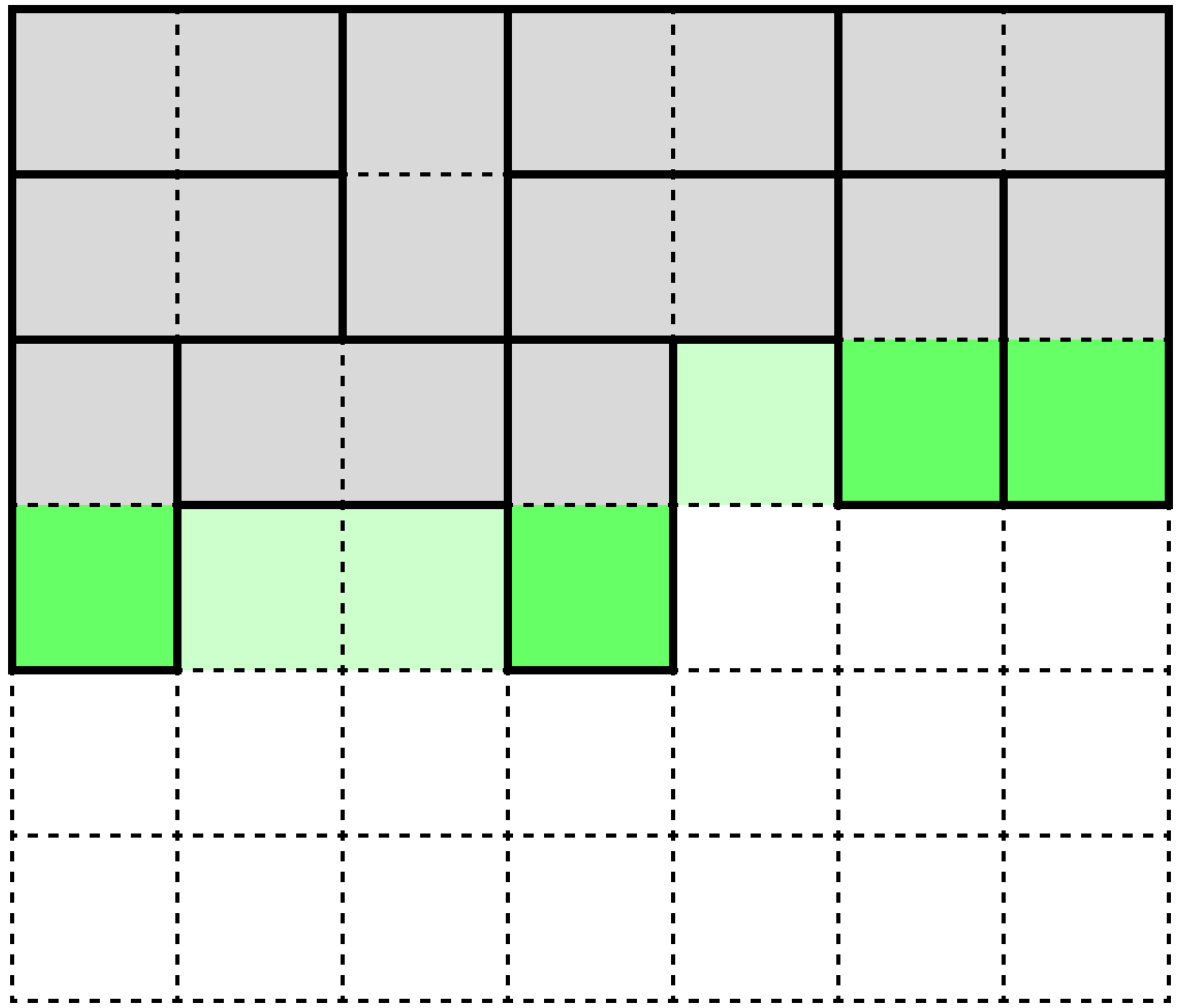
1. i번 칸이 채워져 있는 경우
   - `D[i][state] = D[i+1][state>>1]`
2. i번 칸에 도미노를 수직으로 놓을 수 있는 경우
   - `D[i][state] += D[i+1][state>>1 | 1<<M-1]`



연습문제) BOJ 1648 격자판 채우기

- `D[i][state]` : i번 이전 칸까지 도미노를 놓았고, [i, i+M) 번 칸이 state의 패턴으로 채워져 있을 때, 남은 칸을 채우는 경우의 수

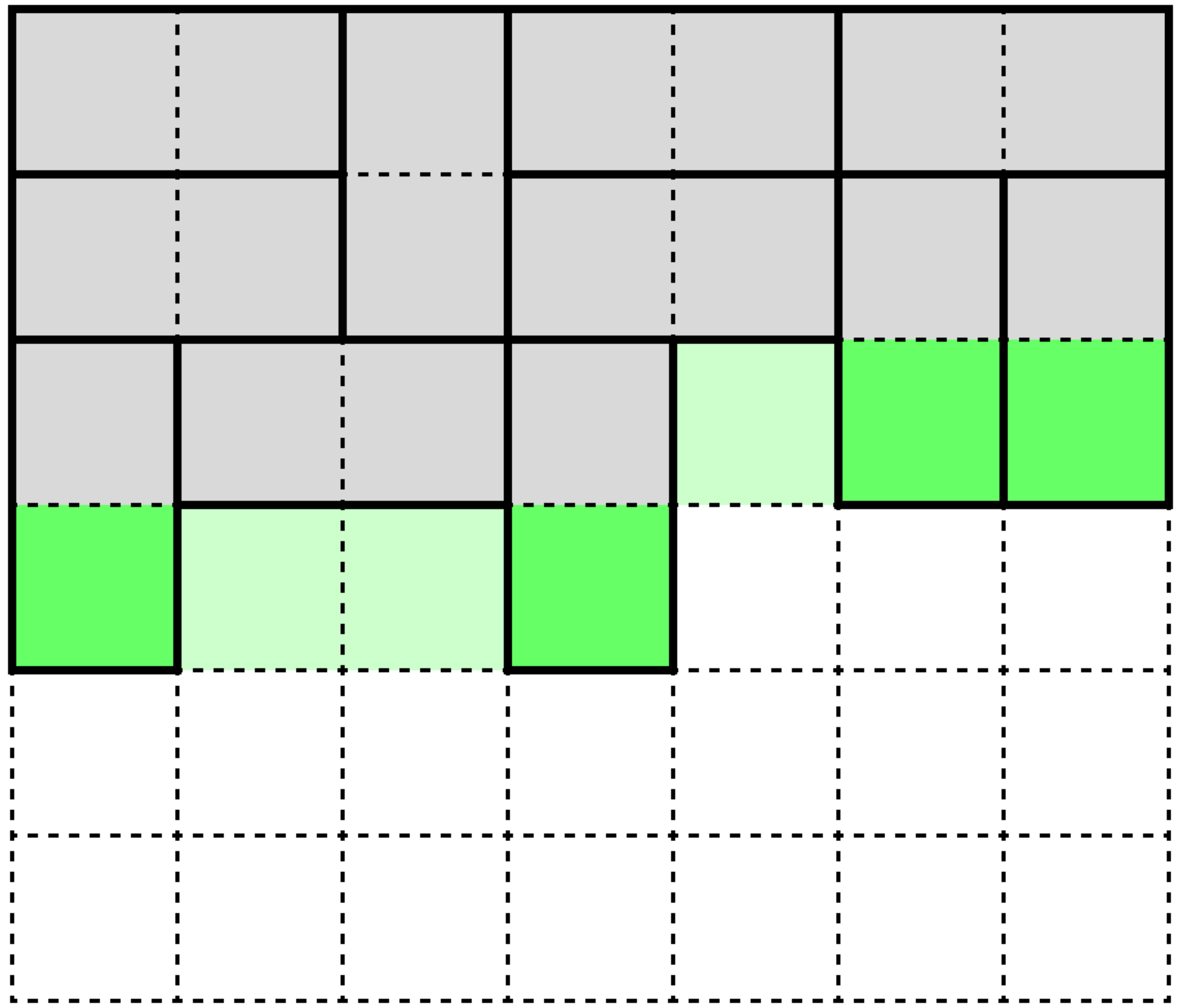
1. i번 칸이 채워져 있는 경우
   - `D[i][state] = D[i+1][state>>1]`
2. i번 칸에 도미노를 수직으로 놓을 수 있는 경우
   - `D[i][state] += D[i+1][state>>1 | 1<<M-1]`
3. i번 칸에 도미노를 수평으로 놓을 수 있는 경우
   - `D[i][state] += D[i+1][state>>1 | 1]`



연습문제) BOJ 1648 격자판 채우기

- `D[i][state]` : i번 이전 칸까지 도미노를 놓았고, [i, i+M) 번 칸이 state의 패턴으로 채워져 있을 때, 남은 칸을 채우는 경우의 수

1. `if (state & 1) // occupied`
   - `D[i][state] = D[i+1][state>>1]`
2. `if (!(state & 1)) // vertical`
   - `D[i][state] += D[i+1][state>>1 | 1<<M-1]`
3. `if (i%M != M-1 && !(state & 3)) // horizontal`
   - `D[i][state] += D[i+1][state>>1 | 1]`

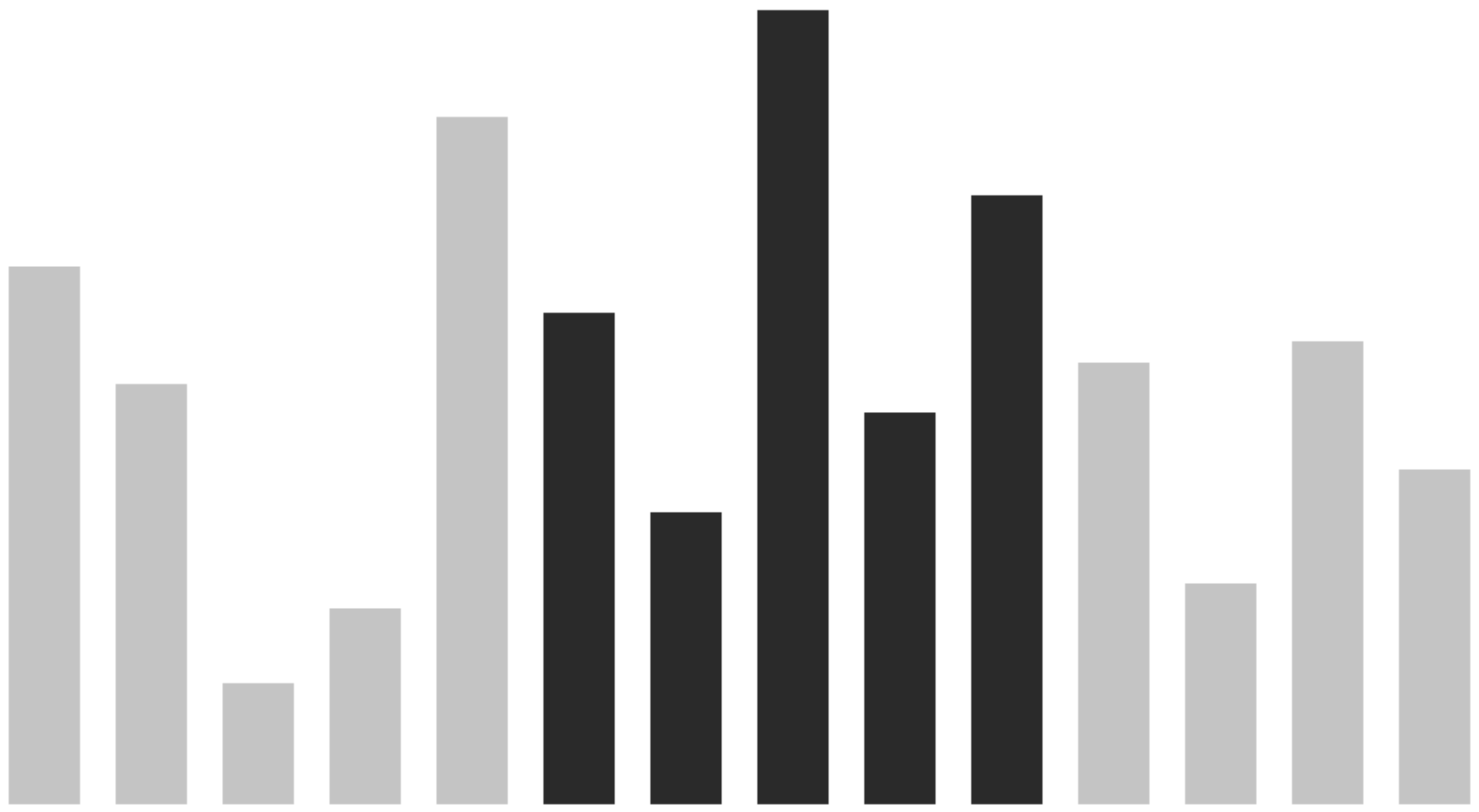


덱 DP

(Deque DP)

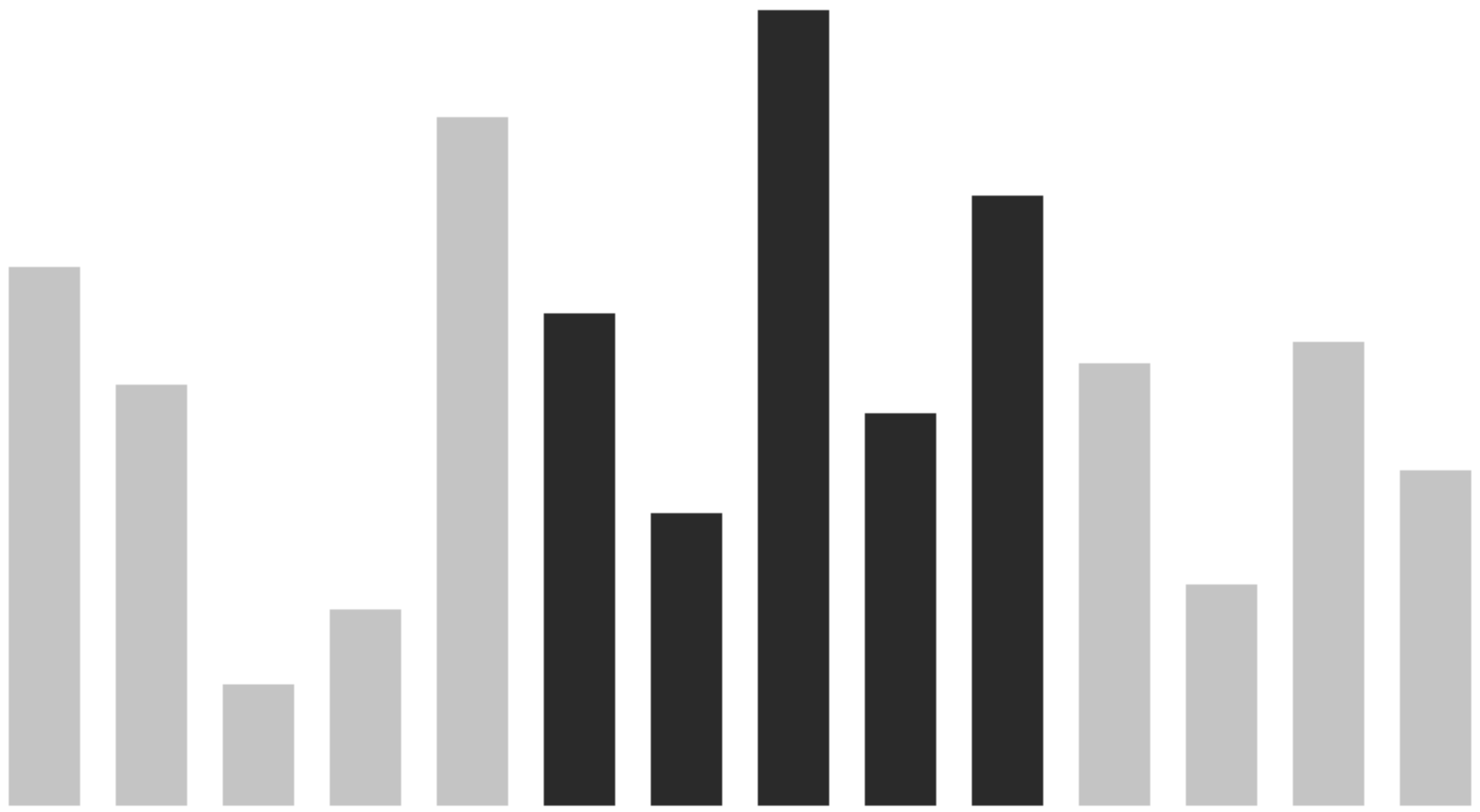
연습문제) BOJ 11003 최솟값 찾기

- $N \le 5{,}000{,}000$ 개의 수 $A_1, \ldots, A_N$이 주어진다.
- 모든 i에 대해, i번째 위치에서 끝나는 길이 L인 구간의 최솟값은?



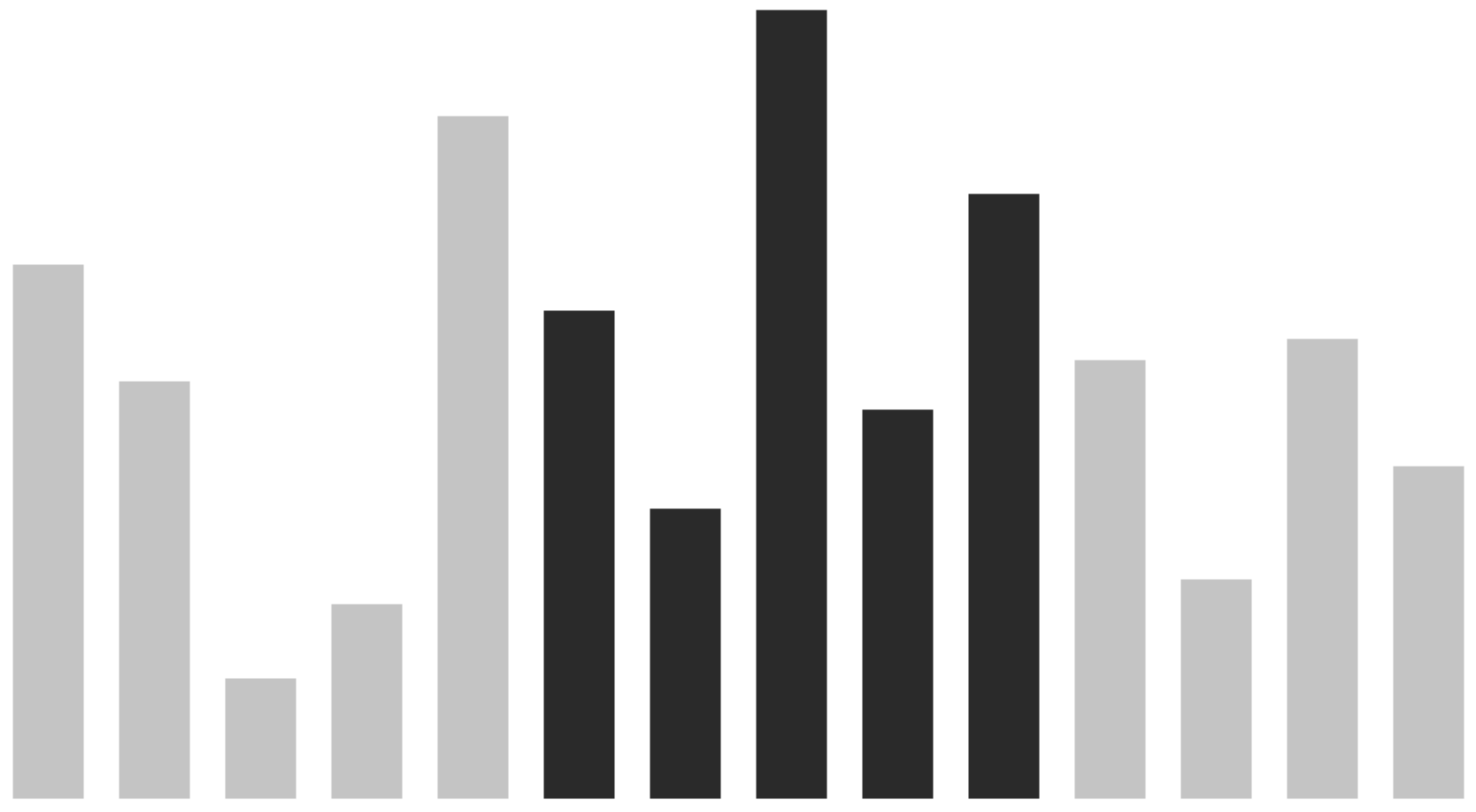
연습문제) BOJ 11003 최솟값 찾기 - 세그먼트 트리

- 가장 간단한 접근: 구간 최솟값 세그먼트 트리
- 시간복잡도: O(N log N)
- N이 크기 때문에 더 빠른 방법을 찾아야 한다!



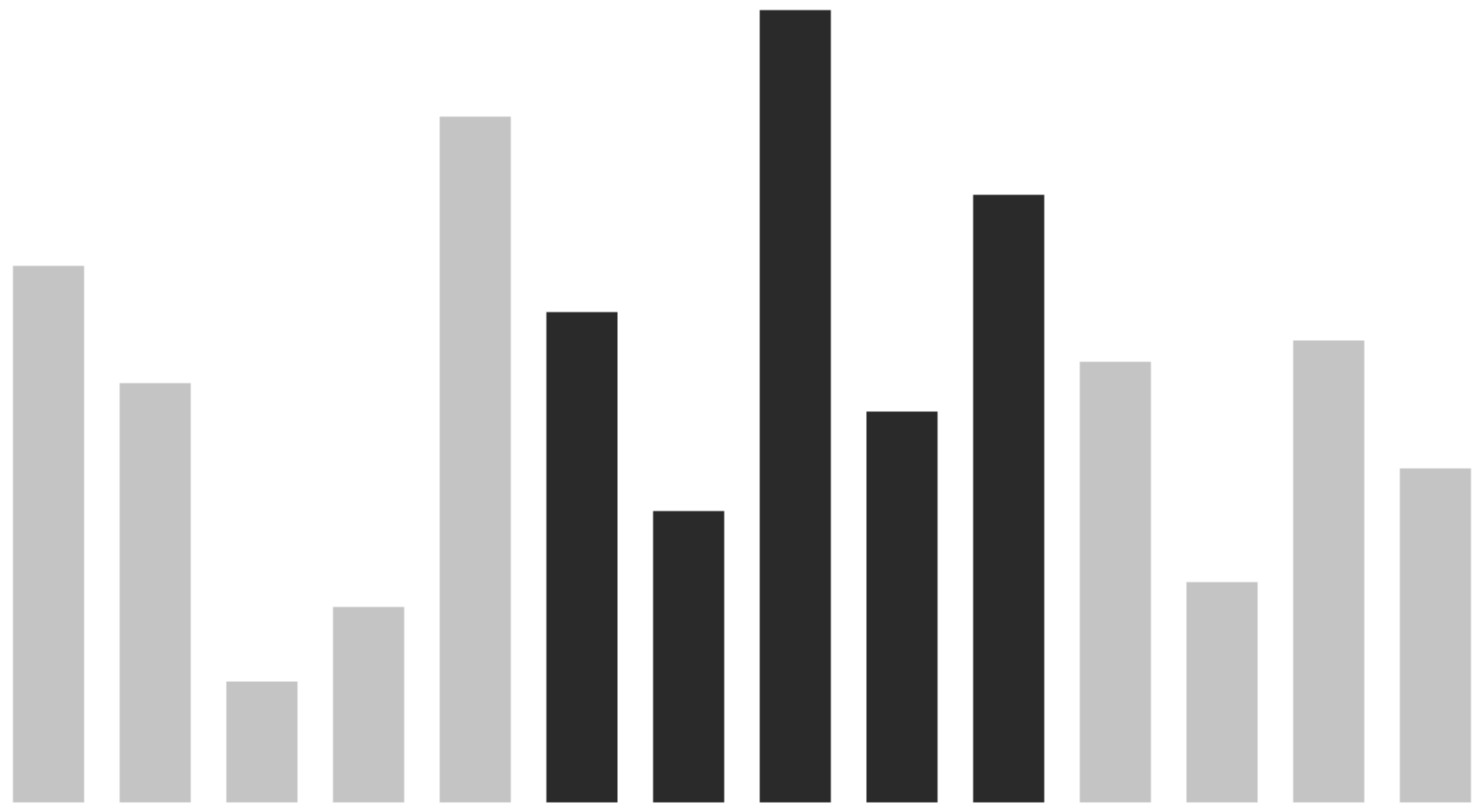
연습문제) BOJ 11003 최솟값 찾기 - 이진 탐색 트리 (set/multiset)

- 구간을 점점 오른쪽으로 옮기면서 구간 안의 원소들을 관리해 보자.
- 구간을 한 칸 옮기면 원소를 하나 추가하고 하나 빼야 한다.
- 집합의 최솟값도 구해야 하므로 순서가 있는 set이나 multiset을 사용한다.
- multiset 원소 추가/제거/최솟값 쿼리: O(log L)
- 시간복잡도: O(N log L)



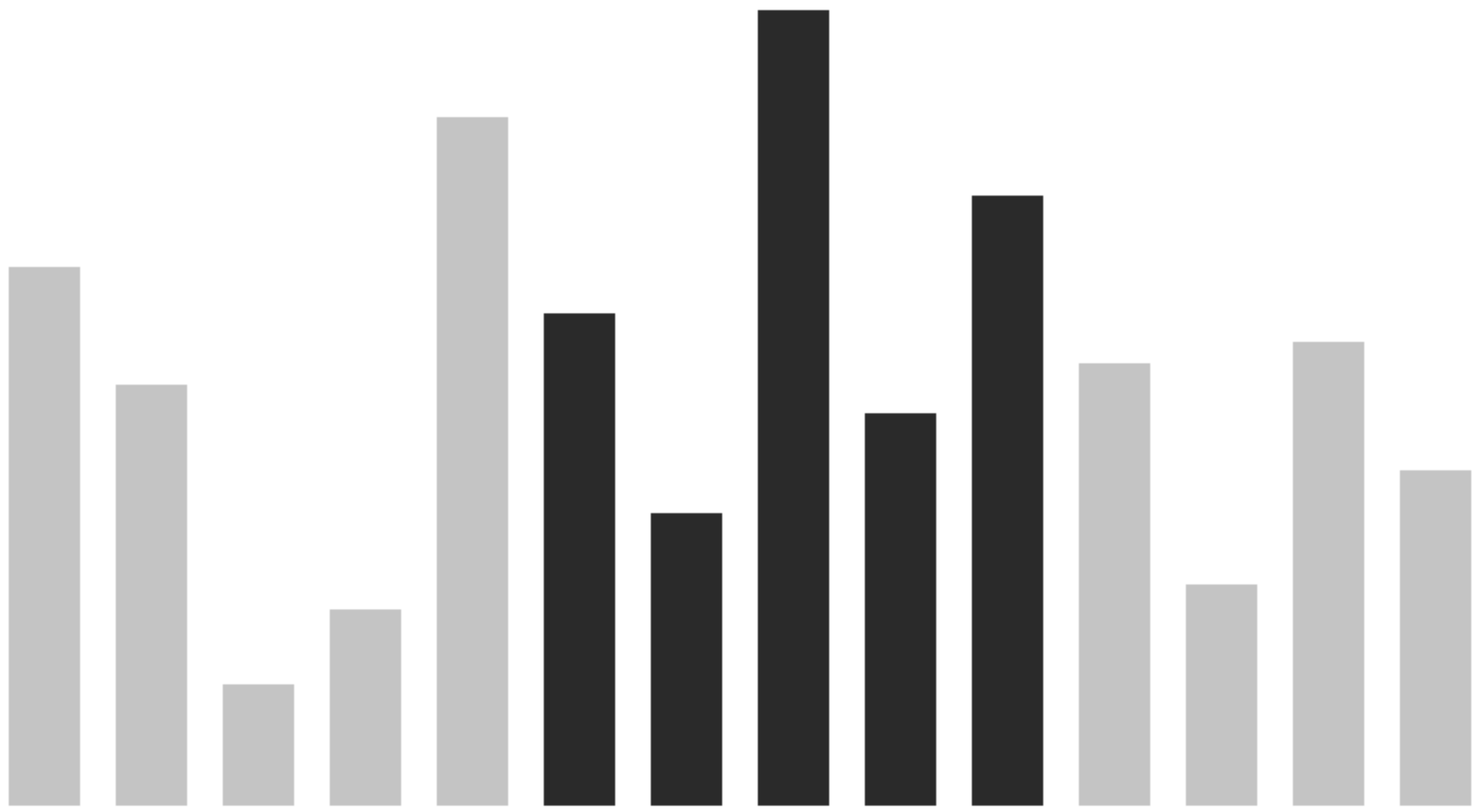
연습문제) BOJ 11003 최솟값 찾기 - 우선순위 큐

- set/multiset보다 빠른 우선순위 큐를 이용할 수는 없을까? (상수 커팅)
- 우선순위 큐에서 원하는 원소를 빠르게 제거하는 것은 어렵다.
- 관찰: 원소를 지금 당장 제거할 필요는 없다! Lazy하게 제거하자.
- 우선순위 큐 안에 있는 최솟값이
    - 구간 안에 있으면 출력
    - 구간 안에 없으면 반복해서 제거



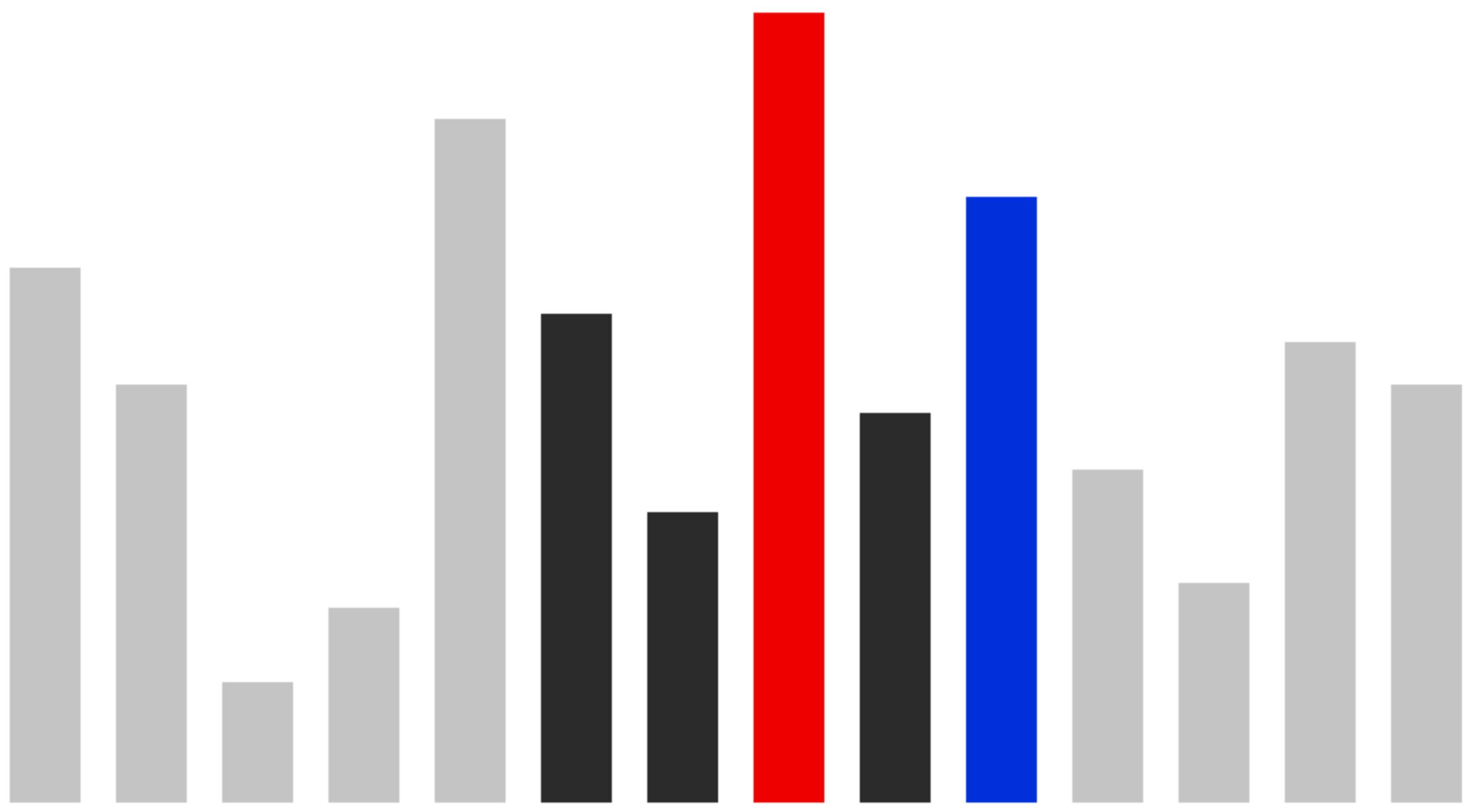
연습문제) BOJ 11003 최솟값 찾기 - 우선순위 큐

- 시간복잡도 분석은?
- 각 원소가 우선순위 큐에 최대 한 번 들어간다.
- 각 원소가 우선순위 큐에서 최대 한 번 빠진다.
    - 한 번 구간에서 빠진 원소는 다시 구간에 포함되지 않기 때문
- 우선순위 큐 원소 추가/제거: O(log N), 최솟값 쿼리: O(1)
- 시간복잡도: O(N log N)



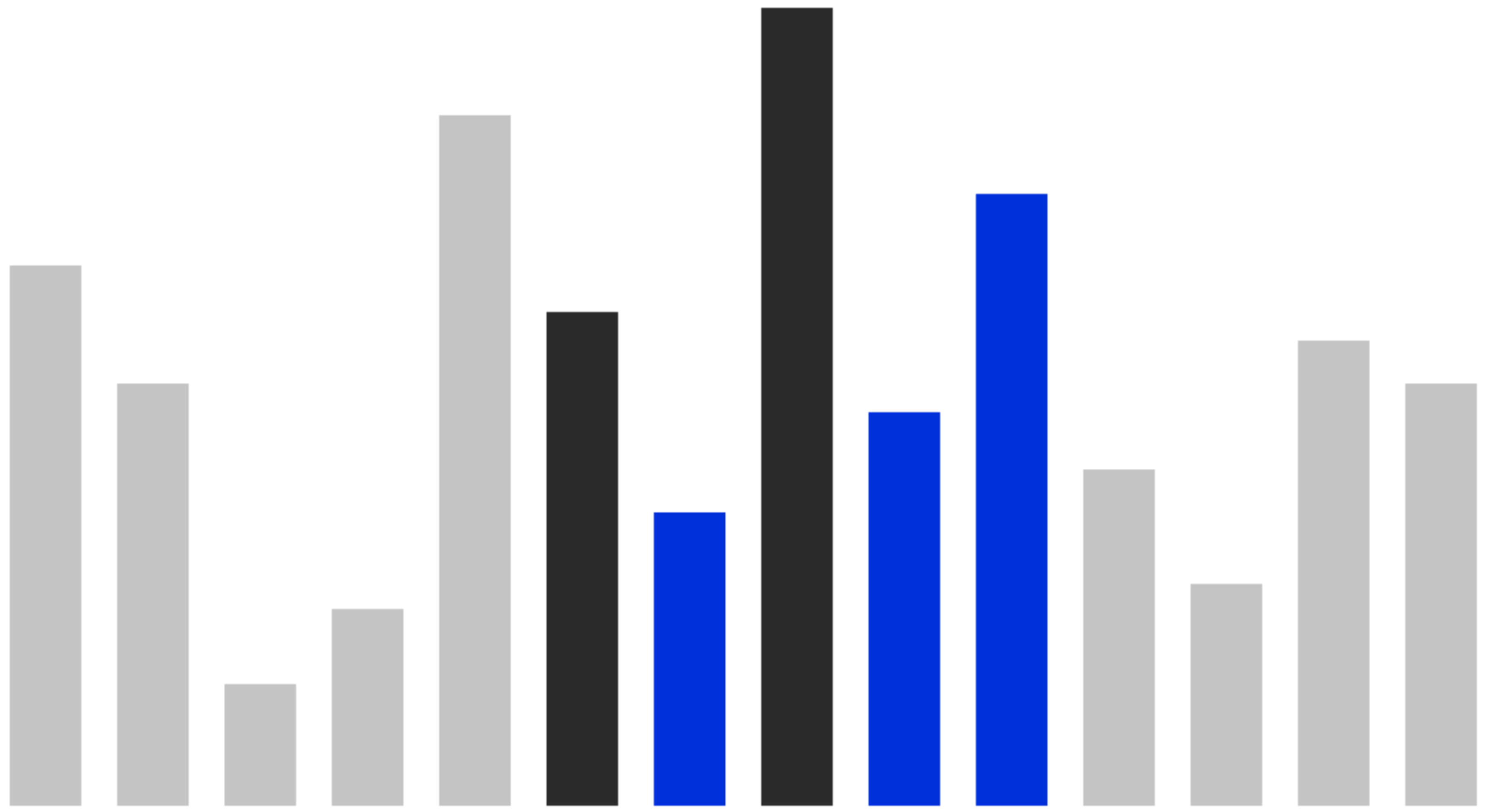
연습문제) BOJ 11003 최솟값 찾기 - 덱

- 관찰: 나중 구간에서 절대 최솟값이 될 수 없는 원소들이 있다.
- $i < j$일 때 A_i가 A_j보다 구간에서 먼저 빠진다. 즉 A_i가 구간 안에 포함된 동안은 A_j도 포함된다.
- $A_i > A_j$ 라면 A_i는 나중 구간에서 최솟값이 될 수 없다.
- $A_i = A_j$ 여도 A_i 대신 A_j를 최솟값으로 잡을 수 있으므로 A_i를 후보에서 제외시킨다.



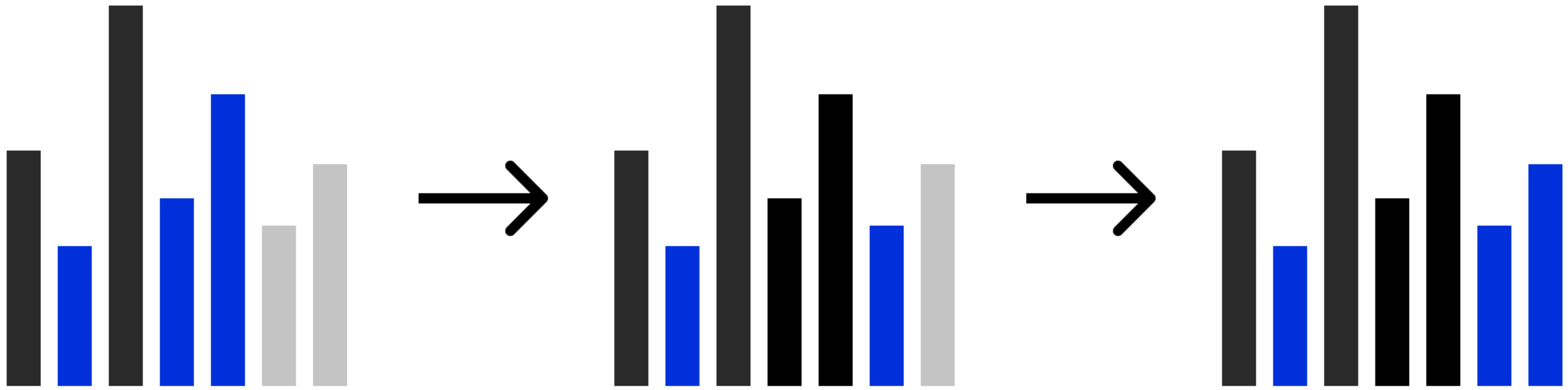
연습문제) BOJ 11003 최솟값 찾기 - 덱

- 최솟값 후보를 관리하면, 인덱스가 증가함에 따라 값이 증가하는 형태가 될 것이다.



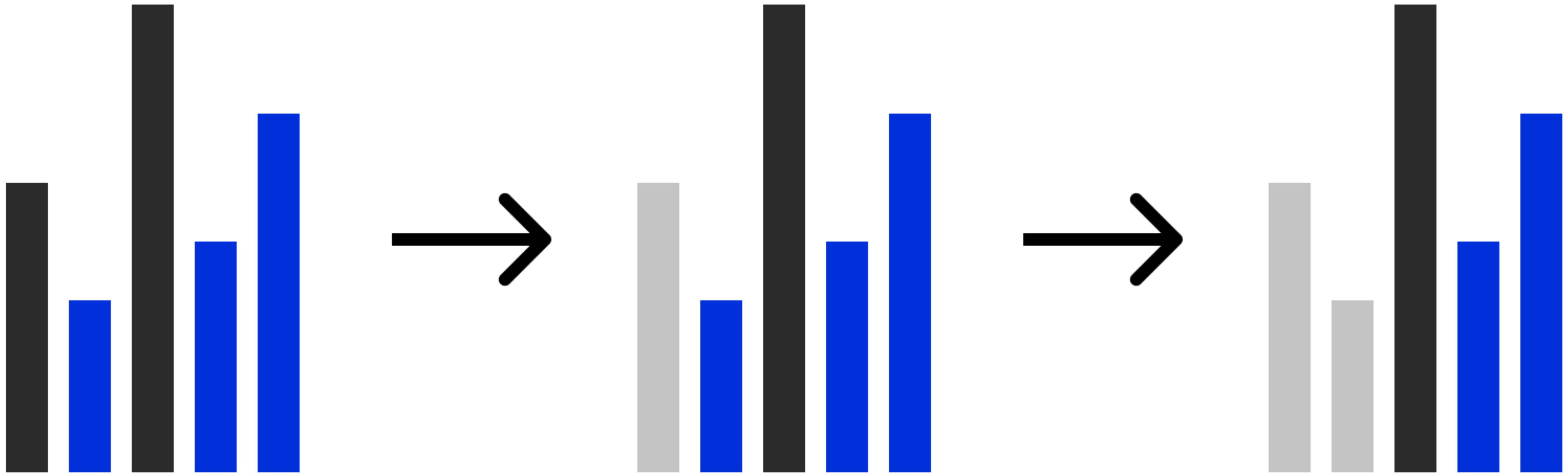
연습문제) BOJ 11003 최솟값 찾기 - 덱

- 최솟값 후보를 관리하면, 인덱스가 증가함에 따라 값이 증가하는 형태가 될 것이다.
- 구간을 오른쪽으로 한 칸 늘릴 때
  - 추가되는 원소 이상인 최솟값 후보는 전부 후보에서 제외시킬 수 있다.



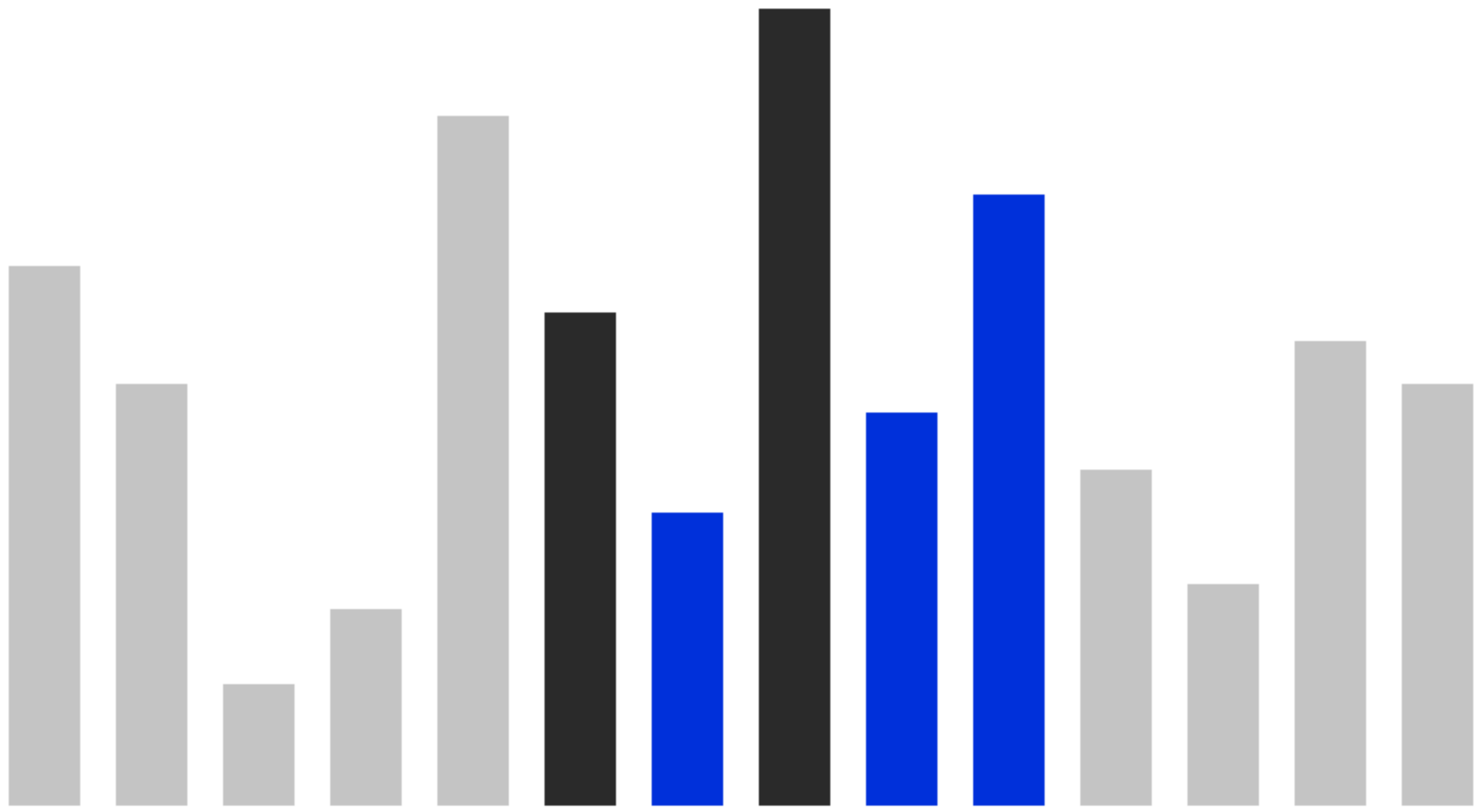
연습문제) BOJ 11003 최솟값 찾기 - 덱

- 최솟값 후보를 관리하면, 인덱스가 증가함에 따라 값이 증가하는 형태가 될 것이다.
- 구간을 오른쪽으로 한 칸 늘릴 때
  - 추가되는 원소 이상인 최솟값 후보는 전부 후보에서 제외시킬 수 있다.
- 구간을 왼쪽에서 한 칸 줄일 때
  - 해당 칸에 최솟값 후보가 있으면 지운다.



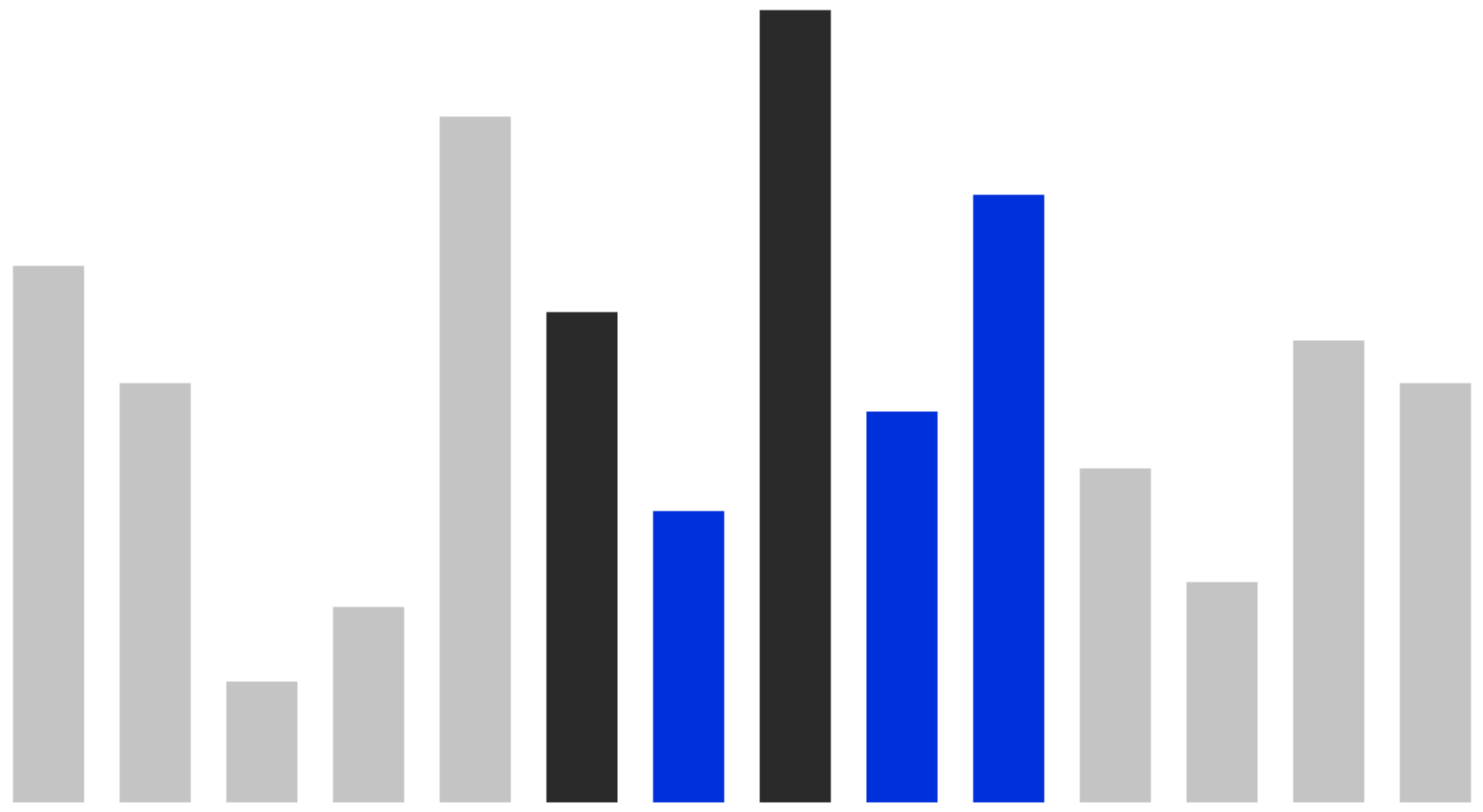
연습문제) BOJ 11003 최솟값 찾기 - 덱

- 최솟값 후보를 관리하면, 인덱스가 증가함에 따라 값이 증가하는 형태가 될 것이다.
- 구간을 오른쪽으로 한 칸 늘릴 때
  - 추가되는 원소 이상인 최솟값 후보는 전부 후보에서 제외시킬 수 있다.
- 구간을 왼쪽에서 한 칸 줄일 때
  - 해당 칸에 최솟값 후보가 있으면 지운다.
- 최솟값 후보 목록의 양 끝에서 원소를 삭제하거나 추가하므로 목록을 덱으로 관리할 수 있다!
- 구간 최솟값 = 덱의 왼쪽 원소



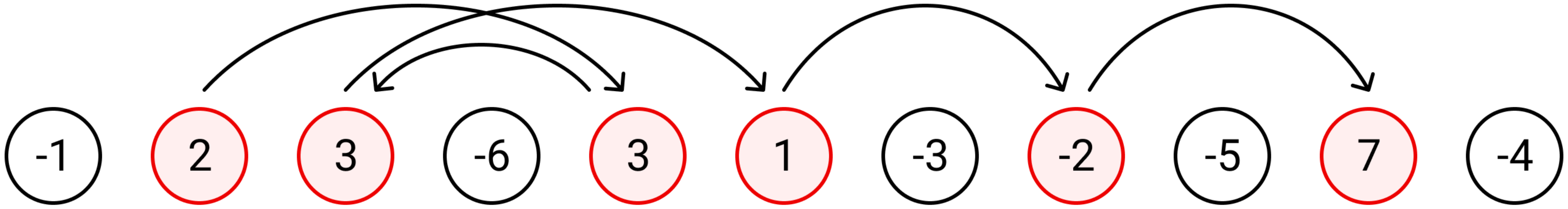
연습문제) BOJ 11003 최솟값 찾기 - 덱

- 시간복잡도 분석은?
- 각 원소가 덱에 최대 한 번 들어간다.
- 각 원소가 덱에서 최대 한 번 빠진다.
  - 새로 추가하는 원소가 작아서 후보에서 탈락하거나,
  - 구간을 왼쪽에서 줄이는 경우
- 덱 원소 추가/제거: O(1), 최솟값 쿼리 (덱의 왼쪽 확인): O(1)
- 시간복잡도: O(N)



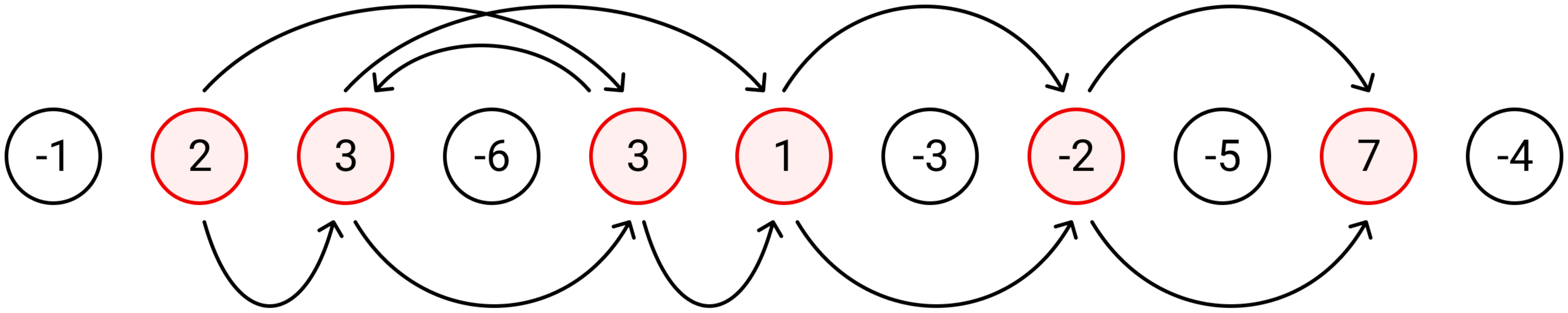
연습문제) BOJ 15678 연세워터파크

- 1부터 $N \le 10^5$ 까지 번호가 붙은 징검다리가 있고, 징검다리마다 정수 K_i가 적혀 있다.
- 원하는 징검다리에서 시작해서 징검다리를 중복 없이 원하는 만큼 밟는다.
- 징검다리 사이를 이동하려면 거리가 D 이하여야 한다.
- 밟는 징검다리에 적힌 정수의 합 최대화



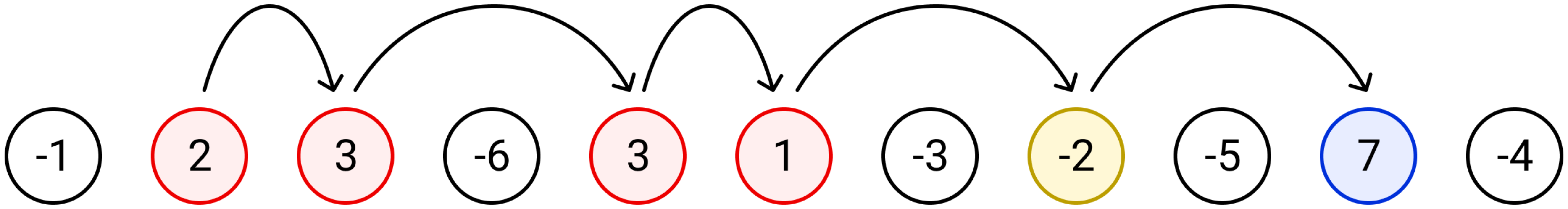
연습문제) BOJ 15678 연세워터파크

- 밟는 순서는 상관이 없고, 밟는 징검다리만 중요하다.
- 관찰: 어떤 징검다리들을 밟는 경로가 존재하면, 이 징검다리들을 왼쪽부터 순서대로 밟을 수 있다.
- 이웃한 징검다리의 거리가 D 이하가 되도록 징검다리를 선택하는 문제로 생각
  - 단순한 DP!



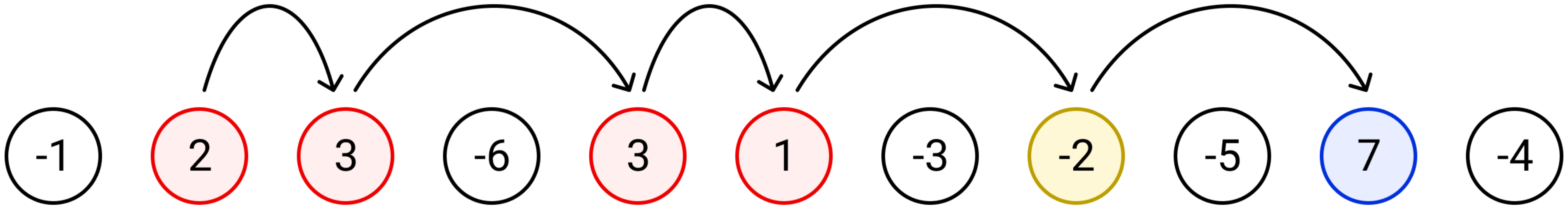
연습문제) BOJ 15678 연세워터파크

- `E[i]` : i번째 징검다리에서 끝나는 경로의 비용
- `E[i] = max(K[i], max_{i-D ≤ j < i} E[j] + K[i])`
- 위 식을 그대로 구현하면 시간복잡도 O(ND)



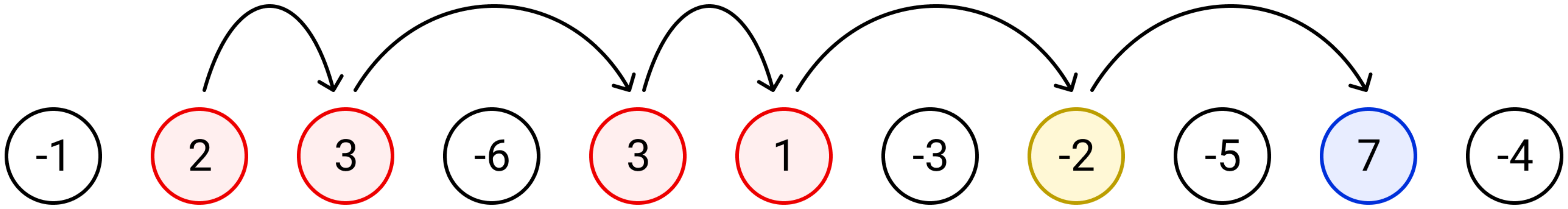
연습문제) BOJ 15678 연세워터파크

- $E[i]$: i번째 징검다리에서 끝나는 경로의 비용
- $E[i] = \max(K[i], \max_{i-D \le j < i} E[j] + K[i])$
- 위 식을 그대로 구현하면 시간복잡도 O(ND)
- 관찰: $E[j]$를 구간 [i-D, i) 에서 선택한다. 덱을 사용하여 [i-D, i) 범위의 E 최댓값을 관리하자!
- 앞 문제와 다른 점: 덱에 추가되는 값이 직접 주어지는 대신 그때그때 계산된다.
- 시간복잡도: O(N)



연습문제) BOJ 15678 연세워터파크

- N 범위가 엄청나게 크지 않으므로 O(N log N) 풀이도 통과된다.
- 세그먼트 트리, set/multiset, 우선순위 큐 구현도 심심하면 짜 보자.



연습 문제

- 2098 - 외판원 순회
- 1648 - 격자판 채우기
- 11003 - 최솟값 찾기
- 15678 - 연세워터파크

출석 문제

- 1562 - 계단 수
- 1086 - 박성원
- 14700 - 넴모넴모 (Hard)
- 14390 - 타일 놓기

- 8201 - Pilots
- 20314 - 대홍수
- 5977 - Mowing the Lawn
- 10129 - 작은 새
- 14869 - 요리 강좌