

예시)

1	2	3
2	3	4
3	4	5

탐색 시작점: (0,0)

탐색 종료점: (3,3) → (N-1, N-1)

delta = [(0,1), (1,0)]

i) 탐색 시작점부터 delta로 이동 (값을 더하면서)

ii) 탐색 종료지점 도달 → 반복 종료

iii) 탐색하면서 더한 값이 최소값인지 확인한다

\*부록 경로의 개수: 예시의 경우 → 2개, ↓ 2개가 필요하다

$\checkmark \downarrow \checkmark \downarrow \checkmark$  ∴ 3H2  
 →자리 →자리 →자리

## [SWEA] 5189 전자카드

예시) N = 3

```
arr =
```

	1	2	3	
0	18	34		1
48	0	55		2
18	7	0		3



0	1	2	3	
0	0	0	0	0
0	0	18	34	1
0	48	0	55	2
0	18	7	0	3

i) 순열 생성 → {1,2,3,1} 1은 고정! ∴ 2,3의 순열을 구하면 된다  
 {1,3,2,1}

• 순열 = [1]

순열에 추가할 값인 2,3을 순회한다

for i in range(2,4)  
 used를 활용하여 포함여부 확인

포함 안되었으면,  
 used[i] = 1 ; 순열.append(i)  
 재귀(k+1, n)  
 used[i] = 0 ; 순열.pop()

순열에 추가하지 않았음을 표시

• used = 

		0	0
0	1	2	3

• 종료조건: 순열의 길이가 N일 때,  
 반복종료 후 순열의 마지막 원소인 1을 추가

ii) 생성된 순열에 맞는 값을 저장한다

ex) 순열 = [1,2] + arr[1][2]

↓

순열 = [1,2,3] + arr[2][3]

↓

순열 = [1,2,3,1] + arr[3][1]

더해야 하는 값: arr[순열[-2]][순열[-1]]



예시)  $N = 3$  (컨테이너)  
 $M = 2$  (트럭)

화물 = 

1	5	3
---	---	---

 → 내림차순정렬 

5	3	1
---	---	---

트럭 = 

8	3
---	---

 → 내림차순정렬 

8	3
---	---

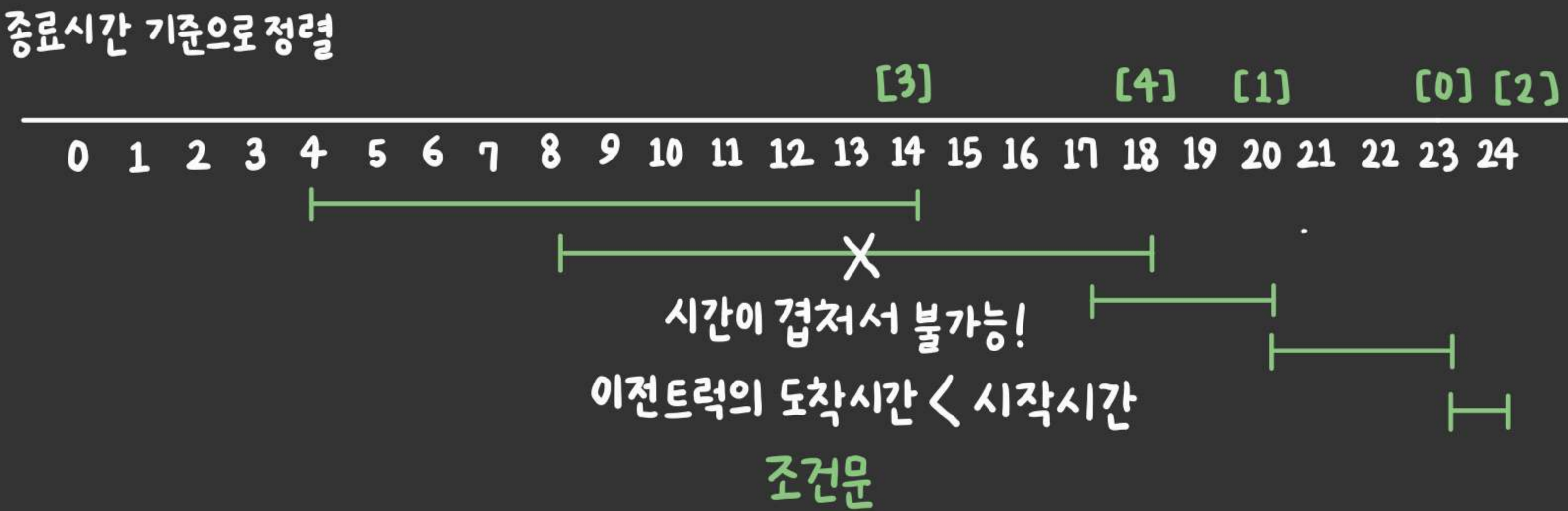
매칭! (한번에 한 화물)

[SWEA] 5202 화물도크

예시)  $N = 5$

도크 = 

20	23
17	20
23	24
4	14
8	18



[BOJ] 9461 파도반 순열

예시)  $N = 11$

