주최 아주대학교 AJOU UNIVERSITY | SW중심대학사업 주관 ANSI 후원 Samsung Software Membership SOLVED.AC STARTLINK

Official 공식 해설 Solutions

Ajou Univ. 2025 아주대학교 프로그래밍 경시대회 Programming Contest

Staff

APC '25

운영

- 김현빈
- 이종학

출제

김현빈 akim9905	아주대
권도현 silvester71	아주대
유상혁 golazcc83	아주대 졸업생
이종학 js9028	아주대
장민우 pani	아주대

검수

jthis	국민대
lky7674	삼성전자
bnb2011	DGIST
mujigae	서울대
ystaeyoon113	연세대
tlsdydaud1	한양대

Div. 1	Div. 2	문제	난이도	출제자
-	**A**	Acentuación del idioma español	**Easy**	권도현
-	**B**	클리크조절	**Easy**	김현빈
A	**C**	마지막 수강신청	**Easy**	권도현
B	**D**	카드 뭉치	**Normal**	유상혁
C	**E**	Was It a Cat I Saw	**Normal**	김현빈
D	**F**	가오리 그래프	**Normal**	유상혁
E	**-**	공중화장실 공리주의	**Normal**	권도현
F	**G**	양과 늑대	**Hard**	유상혁
G	**-**	구간 단속 종료 지점입니다	**Hard**	이종학
H	**-**	2025 만들기	**Hard**	장민우
I	**-**	박지성 카페 다녀왔습니다	**Hard**	이종학

-/2A. Acentuación del idioma español

`implementation`

출제진 의도 – **Easy**

- 출제자: 권도현 silvester71

- 지문에 주어진 **규칙적 강세** 규칙을 그대로 구현하면 됩니다.
- 강세를 적용받을 모음을 찾았다면 해당 모음의 위치를, 찾지 못했다면 `-1`을 출력합니다.
- 실제 스페인어의 발음을 배우고 싶으시다면 : CGEX202 스페인어 (Spanish Language)

-/2B. 클리크 조절

math

출제진 의도 – **Easy**

- 출제자: 김현빈[akim9905]

APC '25

- 두 번의 사격의 표적지가 가진 위상이 동일합니다.
- 동일한 위치 관계의 점을 비교하여 답을 구할 수 있습니다.
  - 예) 가장 왼쪽 위 점
- 시간 복잡도는 $\mathcal{O}(N)$ 혹은 정렬을 사용한다면 $\mathcal{O}(N \log N)$ 입니다.

1A/2C. 마지막 수강신청

brute force

출제진 의도 – **Easy**

- 출제자: 권도현[silvester71]

- 후보 과목의 수, N 이 최대 10 이라는 점을 주목해봅시다.
- 수강할 과목 집합을 S 라고 했을 때, S 를 구성할 수 있는 경우의 수는 최대 2^N 임을 알 수 있습니다.
- $N = 10$ 일 때에도, S 를 구성할 수 있는 경우의 수는 최대 1024 가지밖에 되지 않습니다.

- 모든 가능한 S 마다 다음을 확인합니다.
  - 포함된 다른 강의들끼리 겹치는 시간이 있는지 확인
  - 이미 수강계획에 포함된 시간대인지 아닌지를 나타내는 $5 \cdot 24$ 칸의 배열을 통해 확인 가능
  - 포함된 강의들끼리 겹치는 시간이 발생한다면 해당 S 는 유효하지 않은 시간표 구성입니다.
  - 만약 겹치는 시간이 없다면 해당 S 로 수강하게 되는 총 학점을 기록합니다.
- 하나의 S 에서라도 수강 총 학점이 M 학점 이상이라면 YES를 출력합니다.

1B/2D. 카드 뭉치

dynamic_programming
출제진 의도 – **Normal**

- 출제자: 유상혁 golazcc83

- DP 점화식을 생각해봅시다.

$$dp[i] = \text{왼쪽에서부터 } i \text{개의 카드를 규칙에 따라 나눴을 때 카드 뭉치의 최소 개수}$$

- $1 \le x \le i$ 이면서 $A_x \ge i - x + 1$ 이면 x 번째 카드부터 i 번째 카드까지의 카드 뭉치로 나눌 수 있으므로,

$$dp[i] = \min_{1 \le x \le i,\ i-x+1 \le A_x} \left(dp[x-1] + 1\right)$$

- 시간 복잡도는 $\mathcal{O}\left(N^2\right)$ 입니다.

1C/2E. Was It a Cat I Saw

ad_hoc, brute-force

출제진 의도 – **Easy**

- 출제자: 김현빈 akim9905

- 당황하지 마세요.
- 어떤 수를 이진수로 표현했을 때, 앞의 15 비트는 약 $[X - 2^{15}, X + 2^{15}]$ 동안 고정됩니다.
- 따라서 $[X - 2^{15}, X + 2^{15}]$ 의 범위 안에 반드시 이진수가 팰린드롬이 되는 수가 존재합니다.
- 시간 복잡도는 $\mathcal{O}\left(T\sqrt{X}\log X\right)$ 입니다.

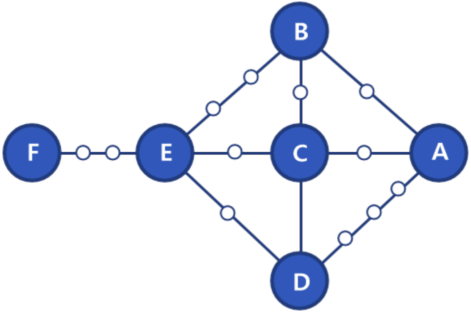
1D/2F. 가오리 그래프

graph_traversal
출제진 의도 – **Normal**

- 출제자: 유상혁[golazcc83]

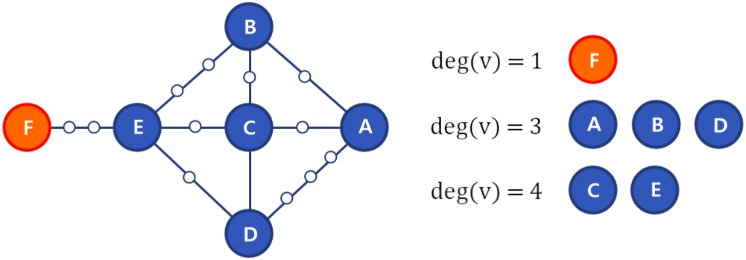
- 가오리 그래프의 가장 큰 특징은 핵심 정점과 일반 정점의 차수(degree, 정점에 인접한 간선의 수) 입니다. 정점 v의 차수를 $\deg(v)$라고 하면,

- $\deg(v) = 1$인 정점은 F로 유일합니다.
- $\deg(v) = 2$인 정점은 일반 정점입니다.
- $\deg(v) = 3$인 정점은 A, B, D입니다.
- $\deg(v) = 4$인 정점은 C, E입니다.



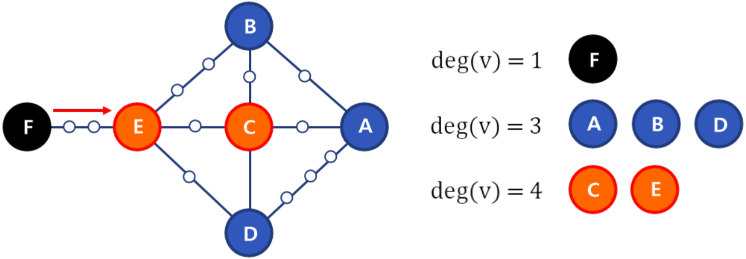
APC '25

- 핵심 정점의 번호를 찾는 방법에는 여러 가지가 있지만, 그 중 하나의 방법을 소개합니다.
- 먼저 $\deg(v) = 1$ 로 유일한 정점 F 의 번호를 확정할 수 있습니다.

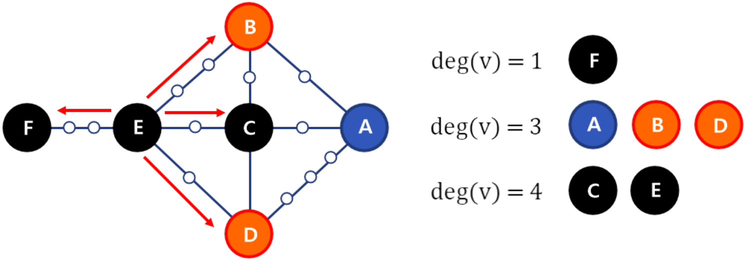


APC '25

- 다음으로, 정점 F 에서 $\deg(v) \neq 2$ 인 정점을 방문할 때까지 그래프 탐색하면 정점 E 의 번호를 찾을 수 있습니다. 또한 $\deg(v) = 4$ 인 정점은 C, E 므로 정점 C 의 번호 또한 자연스럽게 찾을 수 있습니다.



- 다음으로, 정점 E에서 $\deg(v) \neq 2$인 정점을 방문할 때까지 그래프 탐색하면 정점 B, C, D, F를 찾을 수 있습니다. C, F는 이미 번호를 확정한 정점이므로 제외합니다.
- 이 때 정점 B의 번호보다 D의 번호가 더 크므로 둘의 번호 또한 확정할 수 있습니다.
- 마지막으로 $\deg(v) = 3$인 정점 중 아직 방문하지 않은 정점 A의 번호 또한 확정할 수 있습니다.



1E/-. 공중화장실 공리주의

combinatorics, dp

출제진 의도 – **Normal**

- 출제자: 권도현[silvester71]

- $M \le \left\lceil \frac{N}{2} \right\rceil$을 만족하는 경우에는 모두가 심리적 안정감 2를 가질 수 있습니다.
  - 어느 누구도 이웃하지 않아야 하기 때문에, 모든 인원을 우선 다음과 같이 배치합니다.

$$\underbrace{O, X, O, \ldots, X, O}_{2M-1}$$

  - 위 배치에서의 O 사이의 공간 $M+1$개에 남아있는 화장실 칸 $N-(2\cdot M-1)$개를 "끼워넣어" 봅시다.

$$_{M+1}\mathrm{H}_{N-2\cdot M+1}$$

- $M > \left\lceil \frac{N}{2} \right\rceil$ 인 경우에는 모두가 심리적 안정감 2를 가질 수 없습니다.
- 이웃해 있는 사람들을 하나의 그룹으로 생각하면, 한 그룹에서의 심리적 안정감 합은 2입니다.
- 따라서 그룹의 개수를 최대화하는 문제는 심리적 안정감의 총합을 최대화 하는 문제와 동치입니다.

- 마찬가지로, 사람 M 명을 모두 이어서 배치할 것입니다.

$$\underbrace{O, O, O, \dots, O}_{M}$$

- 현재는 그룹이 1 개인 상황입니다. 여기서 X 를 O 사이의 공간에 삽입하여 그룹의 개수를 늘리겠습니다.

$${}_{M-1}\mathrm{C}_{N-M}$$

- 시간 복잡도는 $\mathcal{O}\left(N^2\right)$ 입니다.
  - 이항 계수는 동적 계획법을 이용한다면 $\mathcal{O}\left(N^2\right)$ 에,
  - 페르마의 소정리를 이용한다면 $\mathcal{O}\left(N\right)$ 에 전처리 할 수 있습니다.

1F/2G. 양과 늑대

greedy, two_pointer, binary_search

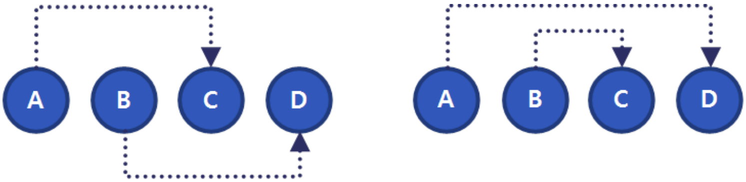
출제진 의도 – **Normal**

- 출제자: 유상혁[golazcc83]

- 양치기 소년이 우리의 개수를 최소로 만들어야 한다면, 동물들을 최대한 많이 매칭시켜야 합니다.
- 또한 이 문제에서는 양 무리와 늑대 무리에서 각각 1 마리씩 데려와서 매칭할 수 있습니다.
- 양과 늑대 각 무리에서 동물들을 최대한 많이 매칭시킨 다음, 두 동물 무리에서 크기가 가장 작은 동물을 하나씩 골라야 합니다.

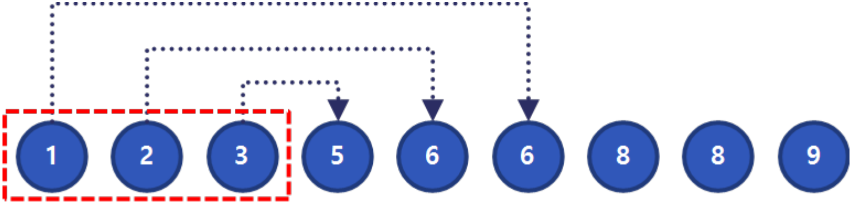
- 먼저 각 동물별로 동물의 크기의 합이 K를 넘지 않도록 최대한 많이 매칭시켜야 합니다.
- 아래의 과정을 통해 매칭시킨 짝의 수를 최대화할 수 있습니다.
  1. 동물을 크기 순으로 정렬
  2. 크기가 작은 동물부터 아직 매칭되지 않은 동물 중 크기가 가장 큰 동물을 선택하여 매칭
- 아래는 $K = 10$일 때의 예시이며, 두 포인터 알고리즘/이분탐색을 사용하여 구현할 수 있습니다.

- 왜 이 방법으로 매칭시키는 것이 최적일까요?
  - 크기가 작은 순으로 동물 A, B, C, D가 있다고 가정합니다.
  - $(A, C), (B, D)$가 매칭 가능하다면, $A + D, B + C \leq B + D$입니다.
  - 따라서 $(A, D), (B, C)$로 다시 매칭시키는 것이 이득입니다.



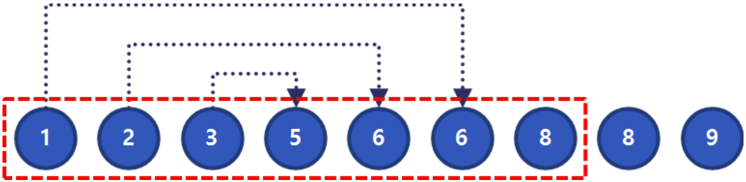
APC '25

- 매칭시킬 동물을 선택할 때에는 크기가 작은 동물을 선택하는 것이 이득입니다.
- 이 때 X 개의 매칭을 만들었다면, 앞의 X 개의 동물은 무조건 작은 쪽으로 매칭됩니다.



APC '25

- 최대 매칭을 유지하면서 다른 동물과의 매칭을 위한 1 마리의 동물을 제외해야 합니다.
- 이 때 제외할 수 있는 후보는 앞에서부터 $2X+1$ 마리의 동물입니다.
- 모든 동물이 매칭되었을 때는 제외할 동물이 없으므로 예외 처리합니다.



- $2X+1$ 개의 동물 중 하나의 동물을 제외했을 때 여전히 X 개의 짝을 만들 수 있는지 판단해야 합니다.

- 차례대로 동물을 하나씩 제외해보면서 X 개의 짝을 만들 수 있게 되는 첫 번째 지점을 찾습니다.
- 동물을 하나를 제외했을 때 X 개의 매칭이 만들 수 있는지는 $\mathcal{O}(N)$ 의 시간복잡도에 판단할 수 있습니다.
- 이분 탐색을 사용하면 $\mathcal{O}(\log N)$ 이므로, $\mathcal{O}(N \log N)$ 의 시간 복잡도에 해당 지점을 찾을 수 있습니다.
- 각 무리에서 위에서 구한 동물의 크기를 더했을 때 K 이하라면, 매칭을 하나 더 만들 수 있습니다.
- 정답은 양과 늑대의 수의 합에서 만들 수 있는 매칭의 수를 제외한 값입니다.
- 시간 복잡도는 $\mathcal{O}(N \log N + M \log M)$ 입니다.

1I/-/-. 구간 단속 종료 지점입니다

sort, sweeping
출제진 의도 – **Hard**

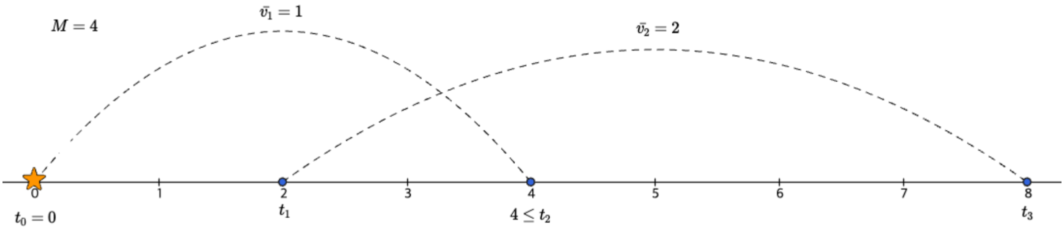
- 출제자: 이종학 js9028

APC '25

- 단속 구간이 하나라면 해당 구간의 평균 속도에 맞춰 이동할 수 있습니다.
- 여러 단속 구간이 겹칠 경우, 과연 가장 낮은 평균 속도에 맞추는 것이 최적일까요?

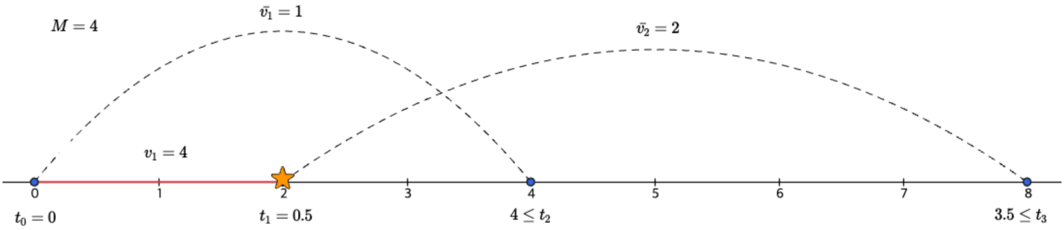
- 평균 속도 $\bar{v} = \dfrac{e-s}{t_2 - t_1}$ 에 따라, $\dfrac{e-s}{\bar{v}} \le t_2 - t_1$ 을 만족해야 합니다.
- 해당 구간을 통과하는 데 필요한 최소 시간은 $\dfrac{e-s}{\bar{v}}$ 입니다.

1G/-. 구간 단속 종료 지점입니다



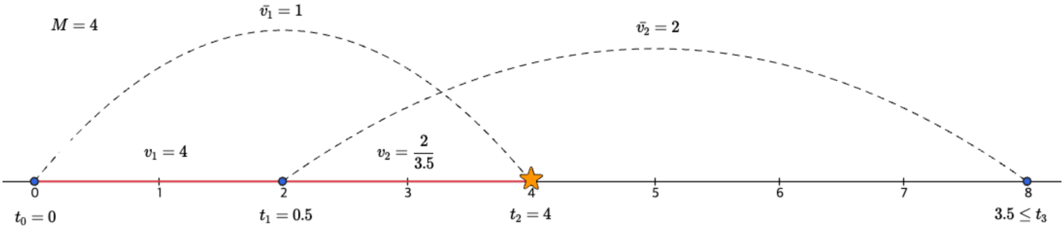
- 단속 구간 1에 대해, 통과에 필요한 최소 시간을 갱신합니다.
- $\dfrac{4}{t_2} \le \bar{v}_1$ 이므로 $4 \le t_2$를 만족해야 합니다.

APC '25



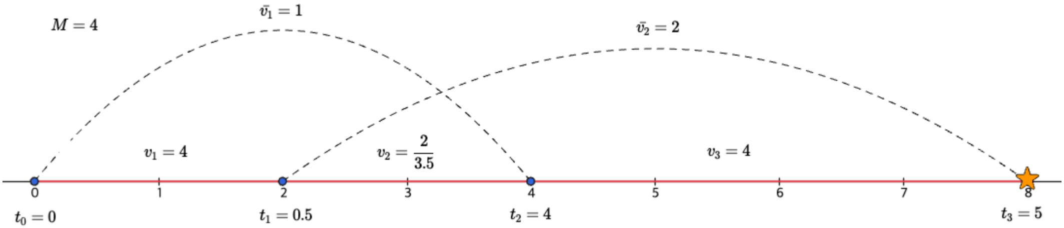
- $x = 2$ 지점에는 최소 시간 제약이 없으므로, 최고 속도인 4로 이동하면 $t_1 = 0.5$에 도달할 수 있습니다.
- 단속 구간 2에서 $\dfrac{6}{t_3 - t_1} \leq \bar{v}_2$ 이므로, $t_3 \geq 3.5$를 만족해야 합니다.

1G/-. 구간 단속 종료 지점입니다



- ✓ $x = 4$에 최고 속도로 도달하면 시간 1에 도착하지만, t_2의 조건을 고려하면 $t_2 = \max(1, 4) = 4$입니다.

1G/-. 구간 단속 종료 지점입니다



- $x = 4$에서 최고 속도로 $x = 8$까지 이동하면 시간 5에 도달하게 되므로, $t_3 = \max(5, 3.5) = 5$입니다.

- 모든 단속 구간의 시작점 s와 끝점 e를 함께 오름차순으로 정렬합니다.
- 항상 최고 속도로 이동한다고 가정하고, 앞에서부터 순서대로 봅니다.
  - s라면 대응되는 위치 e의 통과 가능한 시간을 갱신합니다.
  - e라면 도달하면 통과 가능한 최소 시간을 반영합니다.
- 이는 좌표압축을 통해 해결할 수 있으며, 자료구조 `std::map` 등을 사용해서 해결할 수도 있습니다.
- 정렬에 $\mathcal{O}(N \log N)$, 한번의 스위핑에 $\mathcal{O}(N)$이 소요되어 전체 시간 복잡도는 $\mathcal{O}(N \log N)$입니다.

1H/-. 2025 만들기

`precomputation, constructive`

출제진 의도 – **Hard**

- 출제자: 장민우 pani

- 인접한 두 수 K 와 $K-1$ 에 대하여, $(K-(K-1))$ 의 연산을 통해 1 로 만들 수 있습니다.
- 1 을 어떠한 수에 곱하면, 그 수는 변하지 않습니다.
- 이는, 어떠한 짝수 A 에 대하여 2025를 만드는 구성이 존재한다면, A 이상인 짝수는 전부 구성이 가능하다는 뜻이 됩니다.
- 홀수에 대해서도 마찬가지이므로, 구성이 가능한 가장 작은 짝수와 홀수를 찾으면 문제를 해결할 수 있습니다.

- 우선, N 이 6 이하인 경우 만들 수 있는 가장 큰 수가 2025 보다 작으므로 불가능합니다.
- N 이 7 부터 백트래킹 등의 방법을 이용해 완전탐색을 해보면, N 이 7 일때와 8 일때 모두 2025를 구성 가능함을 알 수 있습니다.
- 백트래킹 코드를 작성하지 않고도, $2025 = 5^2 \times 9^2$ 임을 이용해 손으로도 간단하게 해를 찾을 수 있습니다.
- 여담으로, N 이 8 인 경우 $8 - 4$ 의 연산을 한 번 시행하면, N 이 7 인 경우와 동일해집니다.

1I/-. 박지성 카페 다녀왔습니다

tree, dijkstra

출제진 의도 – **Hard**

- 출제자: 이종학pani, 장민우pani

- 어떤 노드로의 최단 거리가 박지성이 수비수들보다 짧다면, 박지성은 해당 노드에 반드시 도달할 수 있습니다.
- 반대로, 그렇지 않은 노드에는 박지성이 도달할 수 없습니다.
- 따라서, 박지성이 도달할 수 있는 노드만을 대상으로 살펴보면 됩니다.

- 다음의 경우가 존재합니다.
  1. 노드에서 저지당하는 경우
     해당 노드에 수비수가 도달하는 가장 빠른 시간에 박지성이 저지당하게 됩니다.
  2. 간선에서 저지당하는 경우
     간선의 양 끝 노드를 A, B 라고 할 때, 수비수가 이들 노드에 도달하는 가장 빠른 시간을 각각 t_A, t_B, 간선의 가중치를 w 라고 하면, 박지성은 해당 간선 위에서 최대 $\frac{t_A + t_B + w}{2}$ 만큼까지 드리블할 수 있습니다.
- 수비수들의 각 노드까지 최단거리를 구하는데 $\mathcal{O}\left(E \log N\right)$, 답을 구하는데 $\mathcal{O}\left(E\right)$ 만큼 걸립니다.
- 따라서 총 시간복잡도는 $\mathcal{O}\left(E \log N + E\right)$ 입니다.

-/-. 돌아온 밤양갱

sprague-grundy, dp, segment tree

출제진 의도 – **Challenge**

✓ 출제자: 장민우[pani]

- 1 번 쿼리에서 주어지는 고정된 길이 T 의 문자열 S 를 와일드카드 문자열이라고 합시다.
- 쿼리를 처리하기 이전에, 밤양갱 문자열의 입력되어 있는 길이가 x, 와일드카드 문자열의 길이가 y 일때 해당 문자열의 그런디 수를 전처리합니다.
- 나이브하게 생각하면, 모든 (x, y) 쌍에 대해 이동 가능한 게임판의 모든 그런디 수를 가져와서 mex 를 계산하는 방식으로 $O(N'^3)$ 에 계산할 수 있습니다. $(N' = 10 \cdot N + 8)$

APC '25

- 시간복잡도를 줄이기 위해 고정된 x에 대해 y를 변화하면서 생각해봅시다. 만약 y가 N'부터 시작하여 0까지 감소하는 형태라면, 이동 가능한 게임판의 수는 단조감소하는 형태가 됩니다.
- 세그먼트 트리에 다음 상태로 가능한 그런디 수가 등장하는 최소 인덱스를 업데이트 한다면, 어떠한 (x, y) 쌍에 대해 이동 가능한 최대 위치가 (x', y) 라고 했을 때 등장하는 최소 인덱스가 x'를 초과하는 가장 작은 그런디 수를 구하면 해당 값이 mex가 되고, 이는 walking on segtree 등으로 구할 수 있습니다.
- 따라서 $O(N'^2 logN')$에 모든 (x, y) 쌍에 대한 그런디 수를 전처리한 뒤, 이후 쿼리는 나이브하게 $O(QM)$의 시간복잡도로 계산할 수 있습니다.