나는코더다 2023 송년 대회 풀이

Official Solutions

by

나는코더다 39기

문제		의도한 난이도	출제자
A	물고기 게임	**Easy**	채이환
B	순열의 개수	**Easy**	이동현
C	Split the SSHS 2	**Hard**	박영우
D	순열과 수열	**Medium**	박영우
E	마비노기 가방 정리하기	**Challenging**	문정후
F	트리의 개수	**Medium**	이동현
G	금고 털이 2	**Hard**	박영우
H	초전도체 부수기	**Easy**	채이환
I	순찰 업무	**Hard**	문정후
J	줌	**Challenging**	박영우
K	카드 색칠 2	**Medium**	박영우
L	정기 모임 5	**Easy**	이동현

A. 물고기 게임

`case-work`

출제 의도 – **Easy**

- 제출 77번, 정답 14명 (정답률 31%)
- 처음 푼 팀: 作別이다, 最強이여, 26분
- 출제자: 채이환

A. 물고기 게임

Special Thanks to ...

$2 \nmid N$ 인 경우

- 오토를 기준으로 생각합니다.
- $K = A_{1,1} + \cdots + A_{1,[N/2]} + A_{2,1} + \cdots + A_{2,[N/2]}$, $X = A_{1,1} + \cdots + A_{1,N}$, $Y = A_{1,1} + \cdots + A_{1,[N/2]+1} + A_{2,1} + \cdots + A_{2,[N/2]+1}$ 라 합시다.
- 오른쪽으로 가다가 아래로 가서 다시 왼쪽으로 가는 방법으로 K 만큼은 확실히 얻을 수 있습니다.
- 오토는 데이브의 행동에 의해 $\min(X, Y)$ 만큼은 항상 얻을 수 있습니다. 가운데 지점에서 데이브에게 선택권이 있습니다.
- 따라서 오토는 $\max(K, \min(X, Y))$ 만큼 가져가는 것이 최선입니다.

$2|N$ 인 경우

- 데이브를 기준으로 생각합니다.
- $K = A_{1,N/2+2} + \cdots + A_{1,N} + A_{2,N/2+2} + \cdots + A_{2,N}, X = A_{2,1} + \cdots + A_{2,N}, Y = A_{1,N/2+1} + \cdots + A_{1,N} + A_{2,N/2+1} + \cdots + A_{2,N}$ 라 합시다.
- 왼쪽으로 가다가 위로 가서 다시 오른쪽으로 가는 방법으로 K 만큼은 확실히 얻을 수 있습니다.
- 데이브는 오토의 행동에 의해 $\min(X, Y)$ 만큼은 항상 얻을 수 있습니다. 가운데 지점에서 오토에게 선택권이 있습니다.
- 따라서 데이브는 $\max(K, \min(X, Y))$ 만큼 가져가는 것이 최선입니다.

B. 순열의 개수

ad-hoc

출제 의도 – **Easy**

- 제출 34번, 정답 5팀 (정답률 15%)
- 처음 푼 팀: 作別이다, 最強이여, 57분
- 출제자: 이동현

B. 순열의 개수

- 고른 수 중 최댓값이 A에 속한 경우의 수를 구합시다.
- B에 속한 경우의 수는 같은 방법으로 구하면 됩니다.
- $i > 0$를 고정했을 때, 가능한 j가 존재하는지 판단하면 됩니다.
- 각 i에 대해 가능한 j는 최대 하나입니다.

B. 순열의 개수

- $M = \max(A_1, \cdots, A_i)$ 라 합시다.
- 가능한 j 값의 후보는 $M - i + 1$ 밖에 없습니다.
- 실제로 가능한지는 값이 겹치지 않는지, $\max(B_1, \cdots, B_{M-i+1}) < M$ 인지 보아 판단할 수 있습니다.
- 시간 복잡도는 $\mathcal{O}(N)$ 입니다.

C. Split the SSHS 2

`dfs, case-work`

출제 의도 – **Hard**

- 제출 8번, 정답 0팀 (정답률 0%)
- 처음 푼 팀: 없음
- 출제자: 박영우

C. Split the SSHS 2

- 트리에서 임의의 정점을 루트로 정하고, dfs를 시행하여 dfs 트리를 만듭시다.
- $\{a, b, c\}$ 를 골라 이들 정점을 연결하는 간선을 제거한다면 최대 세 개의 간선이 제거됩니다.
- 집합 $\{a, b, c\}$ 을 골랐을 때 제거되는 간선 중 dfs 트리에서의 트리엣지의 개수로 경우를 나눠 생각합니다.
- 트리엣지는 스패닝트리를 형성하므로, 최대 두 개만이 트리엣지입니다.
- 트리엣지가 제거되지 않는 경우, 그래프가 분할되지 않습니다. 따라서 이 경우는 고려할 필요가 없습니다.

C. Split the SSHS 2

제거되는 트리 엣지가 1개

- 트리엣지를 (a, b)라 합시다. a가 부모, b가 자식 정점이라 합시다.
- (a, b)가 단절선일 경우, 어느 c를 골라도 됩니다. 이때, 제거되는 트리 엣지가 1개여야 하므로 a, b는 c와 트리엣지로 인접하지 않아야 합니다.
- 단절선이 아닐 경우, b의 자손 정점과 아닌 정점을 잇는 백엣지가 최대 하나여야 하고, 만약 존재한다면 a나 b중 하나의 정점과 연결되어 있어야 합니다. 전처리를 통해 각 트리엣지 당 $O(\log M)$에 풀 수 있습니다.

제거되는 트리 엣지가 2개

- 트리엣지 상에서 한 정점이 다른 두 개의 정점과 인접합니다. 이 정점을 a라 합시다.
- b, c가 모두 a의 자식인 경우, 둘 중 하나만 a의 자식이고 다른 하나는 a의 부모인 경우가 있습니다.

제거되는 트리 엣지가 2개

- b, c가 모두 a의 자식인 경우, b와 c 사이에는 간선이 있을 수 없습니다. 따라서 제거되는 간선은 $(a, b), (a, c)$ 총 두 개입니다.
- 제거되는 두 간선 중 하나가 단절점이라면 그래프는 분열됩니다. 그렇지 않다면 분열되지 않습니다.
- 각 a에 대해 $O(deg(a))$ 시간에 모든 $\{a, b, c\}$ 쌍을 전부 확인할 수 있습니다.

C. Split the SSHS 2

제거되는 트리 엣지가 2개

- b가 a의 부모, c가 a의 자식인 경우는 첫 번째 케이스와 비슷한 방식으로 처리할 수 있습니다.
- dfs 트리의 트리엣지만 고려할 때, 간선을 제거하면 그래프는 총 세 개의 연결 요소로 분할됩니다.
- 각 컴포넌트를 연결하는 백엣지가 존재하는지 확인하면 됩니다. $O(\log M)$에 처리할 수 있습니다.
- 각 a에 대해 총 $O(deg(a))$개의 c만 고려하면 됩니다.

전체 시간복잡도는 $O(N \log M)$입니다.

D. 순열과 수열

ad-hoc

출제 의도 – **Medium**

- 제출 26번, 정답 6팀 (정답률 27%)
- 처음 푼 팀: 가끔씩 툭하고 C언어로 부끄러워하는 옆자리의 이성호 양, 124분
- 출제자: 박영우

- 순열 A를 사이클들로 분할합니다.
- 어떤 i에 대하여 B_i가 정해진 경우, $A_{B_i} = B_{A_i}$이므로 A_i의 B값도 정해지게 됩니다. 이처럼 i와 같은 사이클에 포함된 모든 원소의 B값은 정해지게 됩니다.

D. 순열과 수열

- A를 구성하는 두 사이클 p와 q가 있어서 p의 한 원소의 B값이 q의 한 원소가 되는 상황을 고려해봅시다. 즉, $p_0, p_1, ..., p_{n-1}, q_0, q_1, ..., q_{m-1}$이 있어 $A_{p_i} = p_{i+1}, A_{q_i} = q_{i+1}$이라 하고, $p_i = pi + n, q_i = q_{i+m}, B_{p_0} = q_0$인 경우입니다.
- 모든 i에 대해 $B_{p_i} = q_i$가 됩니다. 따라서 n이 m의 배수여야 합니다.

D. 순열과 수열

- 각 사이클에 대하여 그 사이클을 이루는 원소들의 B 값을 독립적으로 정해주면 됩니다.
- 한 사이클에 대하여 크기가 그 사이클의 약수가 되는 모든 사이클들에 대하여 최적해를 구하면 됩니다. 크기 X 인 사이클과 Y 인 사이클에 대해 $O(XY)$ 시간 복잡도에 구할 수 있습니다.
- 사이클들의 크기를 내림차순으로 $a_1, a_2, \cdots a_k$ 라 하면 전체 시간복잡도는 $O(\sum_{i=1}^{k}\sum_{j=1}^{k} a_i a_j) = O((\sum a_i)^2) = O(n^2)$ 입니다.

E. 마비노기 가방 정리하기

`ad-hoc, greedy, parametric-search, segtree, offline-queries`

출제 의도 – **Challenging**

- 제출 8번, 정답 0명 (정답률 0%)
- 처음 푼 팀: 없음
- 출제자: 문정후

E. 마비노기 가방 정리하기

Special Thanks to ...

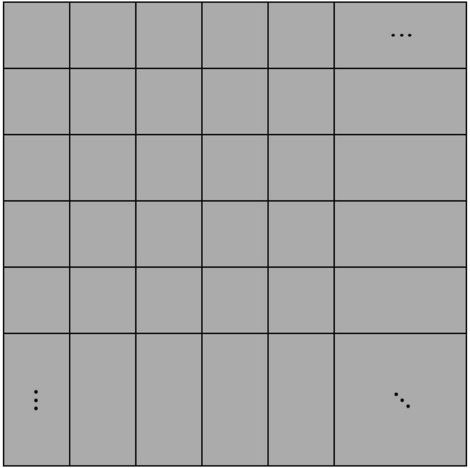
E. 마비노기 가방 정리하기

- 먼저, 가방이 하나라고 생각합시다.
- 가방을 2×2 크기 물건 A 개, 2×1 크기 물건 B 개, 1×2 크기 물건 C 개, 1×1 크기 물건 D 개로 가득 채웠다고 가정합시다.
- 또한, 가방이 가득 차지 않더라도, 가치 0 의 1×1 크기 물건이 무한히 많아서 가득 차게 된다고 생각합니다.
- 가방의 크기는 다음 네 가지 중 하나입니다.
  - 짝수 $\times$ 짝수
  - 짝수 $\times$ 홀수
  - 홀수 $\times$ 짝수
  - 홀수 $\times$ 홀수
- 각 경우에 대해 생각해 봅시다.

E. 마비노기 가방 정리하기

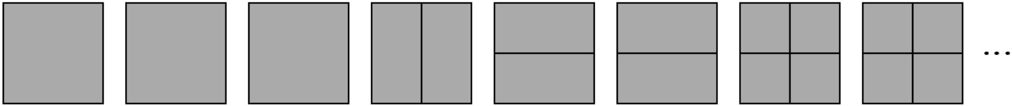
- 가방의 크기가 $R \times C$ 이고 R와 C가 모두 짝수라면, 그 가방은 2×2 크기의 가방이 $\frac{RC}{4}$ 개 있는 것과 동등합니다.
- 이것을 증명하기 위해서는, $R \times C$ 크기의 가방 하나와 2×2 크기의 가방 $\frac{RC}{4}$ 개를 서로 재조합하여 만들 수 있다는 것을 보이면 됩니다.
- 먼저, 2×2 크기의 가방 $\frac{RC}{4}$ 개를 나란히 놓으면 $R \times C$ 크기의 가방을 만들 수 있습니다.
- 가방과 물건들의 넓이 사이의 관계를 생각하면 $4A + 2B + 2C + D = RC$ 입니다.
- R와 C가 모두 짝수이므로, D는 2의 배수이고, $B + C + \frac{D}{2}$ 역시 2의 배수입니다.

E. 마비노기 가방 정리하기



E. 마비노기 가방 정리하기

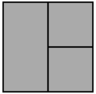
- 2×2 크기의 물건 전부, 2×1 크기의 물건 짝수 개, 1×2 크기의 물건 홀수 개, 그리고 1×1 크기의 물건 4의 배수 개를 사용하여 2×2 크기의 가방을 최대한 많이 채웁니다.



- 2×1 크기의 물건 하나 이하, 1×2 크기의 물건 하나 이하, 2×2 크기의 물건 두 개 이하가 남습니다.

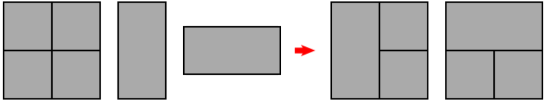
E. 마비노기 가방 정리하기

✓ 넓이 조건을 고려하면 다음과 같은 경우가 있습니다.

남는 2×1 개수	남는 1×2 개수	남는 1×1 개수	설명
0	0	0	
1	0	2	
0	1	2	
1	1	0	*

E. 마비노기 가방 정리하기

- 2×1 크기의 물건과 1×2 크기의 물건이 있는* 경우를 자세히 보겠습니다.
- 만약 1×1 크기의 물건이 하나 이상 사용되었다면, 적어도 네 개가 사용된 것이므로 다음과 같이 2×2 크기의 가방을 채울 수 있습니다.



- 만약 1×1 크기의 물건이 사용되지 않았다면, 애초에 $R \times C$ 크기의 가방을 채우기가 불가능하므로 고려하지 않습니다.
- 그 까닭은 다음과 같습니다.

E. 마비노기 가방 정리하기

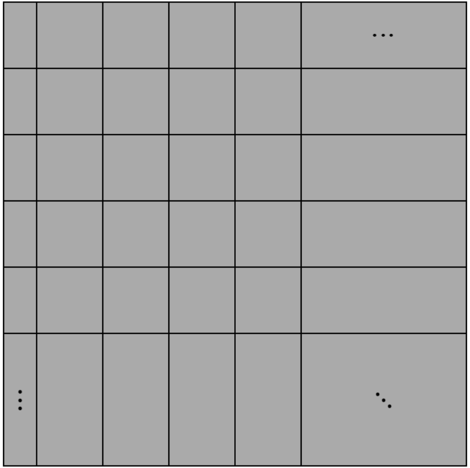
- B와 C가 홀수입니다.
- 이 물건들로 채울 수 있는 격자 (i, j) 중 i가 짝수인 것의 개수는 2×2 크기의 물건마다 두 개, 2×1 크기의 물건마다 한 개, 그리고 1×2 크기의 물건마다 없거나 두 개입니다.
- $2A + B + 2C' = \dfrac{RC}{2}$가 됩니다. C'는 C 이하의 음이 아닌 정수입니다.
- 따라서 B가 짝수이고, 이는 모순입니다.

- 따라서, 모든 경우에서 $R \times C$ 크기의 가방을 2×2 크기의 가방 $\dfrac{RC}{4}$ 개로 나눌 수 있고, 증명이 완료되었습니다.

E. 마비노기 가방 정리하기

- 가방의 크기가 $R \times C$ 이고 R가 짝수, C 가 홀수라면, 그 가방은 2×2 크기의 가방이 $\frac{R(C-1)}{4}$ 개, 2×1 크기의 가방이 $\frac{R}{2}$ 개 있는 것과 동등합니다.
- 먼저, 2×2 크기의 가방 $\frac{R(C-1)}{4}$ 개와, 2×1 크기의 가방 $\frac{R}{2}$ 개를 나란히 놓아서 $R \times C$ 크기의 가방을 만들 수 있습니다.

E. 마비노기 가방 정리하기



E. 마비노기 가방 정리하기

- ✓ 2×1 크기의 가방 $\frac{R}{2}$ 개를 가득 채울 수 있는 $2 \times 1, 1 \times 1$ 크기의 물건이 충분히 존재합니다.
- ✓ 각 행에는 격자가 홀수 개 있습니다. 격자의 개수의 기우성을 바꿀 수 있는 것은 $2 \times 1, 1 \times 1$ 크기의 물건이기 때문에, 각 행에는 $2 \times 1, 1 \times 1$ 크기의 물건의 넓이가 1 이상 있습니다.
- ✓ 따라서 $2 \times 1, 1 \times 1$ 크기의 물건의 넓이의 합은 항상 R 이상이고, 이는 2×1 크기의 가방 $\frac{R}{2}$ 개를 가득 채울 수 있음을 의미합니다.
- ✓ 2×1 크기의 물건으로 2×1 크기의 가방을 채울 수 있는 만큼 채우고, 남은 1×1 크기의 물건을 둘씩 모아 나머지 가방을 채우면 되기 때문입니다.
- ✓ 나머지 2×2 크기의 가방을 채우는 방법은 R, C 가 짝수인 경우와 같습니다.

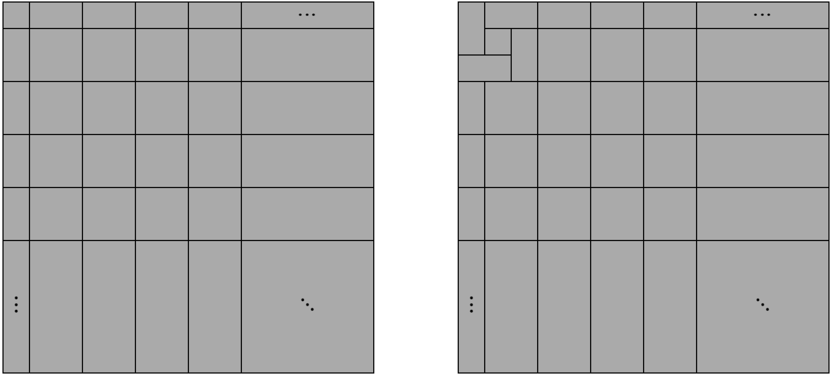
E. 마비노기 가방 정리하기

- 가방의 크기가 $R \times C$ 이고 R가 홀수, C 가 짝수라면, R가 짝수, C 가 홀수인 경우와 대칭적으로 동등합니다.
- 즉, 그 가방은 2×2 크기의 가방이 $\frac{(R-1)C}{4}$ 개, 1×2 크기의 가방이 $\frac{C}{2}$ 개 있는 것과 동등합니다.

E. 마비노기 가방 정리하기

- 마지막으로, 가방의 크기가 $R \times C$ 이고 R, C 가 홀수일 때, $R = 2x + 1, C = 2y + 1$ 이라고 합시다.
- 이때, 기본적으로 그 가방은 2×2 크기의 가방이 xy 개, 2×1 크기의 가방이 x 개, 1×2 크기의 가방이 y 개, 그리고 1×1 크기의 가방이 하나 있는 것과 동등합니다. 그림에서 왼쪽에 해당합니다.
- 또한, $xy > 0$ 인 경우, 그 가방은 2×2 크기의 가방이 $xy - 1$ 개, 2×1 크기의 가방이 $x + 1$ 개, 1×2 크기의 가방이 $y + 1$ 개, 그리고 1×1 크기의 가방이 하나 있는 것과 동등할 수도 있습니다.
- 먼저, 두 경우 모두 가방들을 나란히 놓아서 $R \times C$ 크기의 가방을 만들 수 있습니다.

E. 마비노기 가방 정리하기



E. 마비노기 가방 정리하기

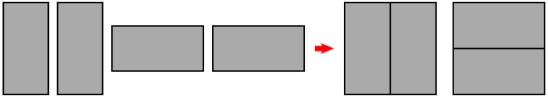
- $(2x+1)(2y+1) = 4A + 2B + 2C + D$가 되므로, D는 홀수입니다.
- 즉, D는 1 이상이고, 따라서 하나의 1×1 크기의 가방은 항상 채울 수 있습니다.
- 또한, 각 행에는 격자가 홀수 개 있습니다. 격자의 개수의 기우성을 바꿀 수 있는 것은 2×1, 1×1 크기의 물건이기 때문에, 각 행에는 $2 \times 1, 1 \times 1$ 크기의 물건의 넓이가 1 이상 있습니다.
- 따라서, $2 \times 1, 1 \times 1$ 크기의 물건의 넓이의 합은 $2x+1$ 이상입니다.
- 이것으로 2×1 크기의 가방 x 개를 모두 채울 수 있습니다.
- 각 열에는 격자가 홀수 개 있습니다. 격자의 개수의 기우성을 바꿀 수 있는 것은 $1 \times 2, 1 \times 1$ 크기의 물건이기 때문에, 각 열에는 $1 \times 2, 1 \times 1$ 크기의 물건의 넓이가 1 이상 있습니다.
- 따라서, $1 \times 2, 1 \times 1$ 크기의 물건의 넓이의 합은 $2y+1$ 이상입니다.
- 이것으로 1×2 크기의 가방 y 개를 모두 채울 수 있습니다.

E. 마비노기 가방 정리하기

- 또한, $R \times C$ 크기의 가방에 2×2 크기의 물건을 최대한 채워 봐도 최소 $2x + 2y + 1$의 넓이가 남고, 이것은 $2 \times 1, 1 \times 2$, 그리고 1×1 크기의 물건으로 가득 채워야 합니다.
- 따라서, $2 \times 1, 1 \times 2$, 그리고 1×1 크기의 물건의 넓이의 합은 $2x + 2y + 1$ 이상입니다.
- 이것은 1×1 크기의 가방 하나, 2×1 크기의 가방 x 개, 그리고 1×2 크기의 가방 y 개를 동시에 채울 수 있음을 의미합니다.
- 나머지 2×2 크기의 가방 xy 개를 채우는 방법은 R, C가 짝수인 경우와 같습니다.
- 다만, 이 경우에는 $xy > 0$인 경우 2×1 크기의 물건 하나와 1×2 크기의 물건 하나가 남는 경우 역시 가능합니다. 그림에서 오른쪽에 해당합니다.
- 2×2 크기의 가방이 $xy - 1$ 개, 2×1 크기의 가방이 $x + 1$ 개, 1×2 크기의 가방이 $y + 1$ 개, 그리고 1×1 크기의 가방이 하나 있는 것과 동등하게 됩니다.
- 따라서 이때에는 두 경우를 모두 고려해야 합니다.

E. 마비노기 가방 정리하기

- 가방이 여러 개인 경우를 생각하면, 가방의 추가, 제거 쿼리마다, $(2,2)$, $(2,1)$, $(1,2)$, $(1,1)$ 크기의 작은 가방 몇 개로 분할되는지를 계산합니다.
- 가방의 추가, 제거 쿼리는 이 네 가지 가방의 개수를 더하고 빼는 연산이 됩니다.
- R, C 가 모두 홀수이고 $xy > 0$ 인 경우 예외가 있었습니다.
- 그러한 예외가 두 개인 경우, 2×1 크기의 물건 두 개, 1×2 크기의 물건 두 개가 되고, 이것은 그림과 같이 2×2 가방 두 개에 담을 수 있습니다.



- 따라서, 예외는 한 번만 고려하여도 충분합니다.

E. 마비노기 가방 정리하기

- 2×2 크기의 가방 P 개, 2×1 크기의 가방 Q 개, 1×2 크기의 가방 S 개, 그리고 1×1 크기의 가방 T 개에 물건을 채우는 문제로 환원되었습니다.
- P, Q, S, T 의 값은 매 쿼리마다 상수 시간 안에 계산할 수 있습니다.
- 먼저, 같은 크기의 물건을 고른다면 가치가 높은 것부터 고르는 것이 이득입니다.
- 따라서, 이 가방들에 $2 \times 2, 2 \times 1, 1 \times 2, 1 \times 1$ 크기의 물건이 각각 몇 개씩 사용해야 최적인지 알면 충분합니다.
- 또한, 2×2 크기의 가방만 채우려고 한다면, 사용한 2×2 크기의 물건 중 가장 가치가 낮은 물건의 가치는 사용한 2×1 크기의 물건 중 가장 가치가 낮은 물건 두 개의 가치의 합과 비슷해야 합니다.
- 그렇지 않다면, 2×1 크기의 물건 두 개 대신 2×2 크기의 물건을 사용하거나, 2×2 크기의 물건 대신 2×1 크기의 물건 두 개를 사용하면 되기 때문입니다.

E. 마비노기 가방 정리하기

- 마찬가지로, 사용한 2×2 크기의 물건 중 가장 가치가 낮은 물건의 가치, 사용한 2×1 크기의 물건 중 가장 가치가 낮은 물건 두 개의 가치의 합, 사용한 1×2 크기의 물건 중 가장 가치가 낮은 물건 두 개의 가치의 합, 사용한 1×1 크기의 물건 중 가장 가치가 낮은 물건 네 개의 가치의 합은 모두 서로 비슷해야 합니다.
- 또한, 2×2 크기의 가방을 채우려고 할 때는 모든 물건을 사용할 수 있지만, 2×1 크기의 가방을 채우려고 할 때에는 $2 \times 1, 1 \times 1$ 크기의 물건만, 1×2 크기의 가방을 채우려고 할 때에는 1×2, 1×1 크기의 물건만 사용할 수 있으며, 1×1 크기의 가방은 오직 1×1 크기의 물건만 사용할 수 있습니다.

E. 마비노기 가방 정리하기

- 따라서, 다음과 같은 탐욕법이 가능합니다.
- 먼저, 1×1 크기의 가방 전부에 1×1 크기의 물건을 가치가 높은 것부터 채울 수 있는 만큼 채웁니다.
- 그다음으로, 2×1 크기와 1×2 크기의 가방 전부에 2×1 크기의 물건을 가치 k 이하, 1×2 크기의 물건을 가치 k 이하로 사용하고, 1×1 크기의 물건 두 개의 가치의 합을 k 이하로 추가 사용하는 k에 대한 이분 탐색을 진행하여, 가치가 높은 것부터 채울 수 있는 만큼 채웁니다.
- 이때, 2×1 크기의 물건과 1×2 크기의 물건은 서로 바꿔 사용할 수 없다는 것에 유의해야 합니다.
- 즉, 2×1 크기의 물건을 2×1 크기의 가방보다 많이 사용하거나, 1×2 크기의 물건을 1×2 크기의 가방보다 많이 사용할 수는 없습니다.

E. 마비노기 가방 정리하기

- 마지막으로, 나머지 2×2 크기의 가방 전부에 2×2 크기 물건을 가치 k 이하로 사용하고, 2×1 크기의 물건 두 개의 가치의 합을 k 이하, 1×2 크기의 물건 두 개의 가치의 합을 k 이하, 그리고 1×1 크기의 물건 네 개의 가치의 합을 k 이하로 추가 사용하는 k에 대한 이분 탐색을 진행하여, 가치가 높은 것부터 채울 수 있는 만큼 채웁니다.
- 이분 탐색 과정에서 서로 다른 두 종류의 물건의 가치의 합이 서로 같은 경우에 유의해야 합니다.
- 이런 탐욕법이 성립하는 이유는, 가방의 크기에 따라 가방에 들어갈 수 있는 물건의 크기가 제한되기 때문입니다.

E. 마비노기 가방 정리하기

- 이것만으로 고려할 수 없는 경우가 생기는데, 그 이유는 다음과 같습니다.
- 2×1 크기의 가방과 1×2 크기의 가방을 채운 뒤 2×2 크기의 가방을 채우는 과정에서, 2×1 크기의 물건 개수의 기우성, 1×2 크기의 물건 개수의 기우성, 그리고 1×1 크기의 물건 개수를 4로 나눈 나머지가 고정되게 됩니다.
- 따라서, 두 과정 사이에서 기우성을 변화시키기 위한 다음과 같은 네 가지 경우를 추가로 처리해야 합니다.

추가 사용하는 2×1 개수	추가 사용하는 1×2 개수	추가 사용하는 1×1 개수	비고
0	0	0	
1	0	2	
0	1	2	
1	1	4	**

E. 마비노기 가방 정리하기

- 2×1 크기의 물건 하나, 1×2 크기의 물건 하나, 그리고 1×1 크기의 물건 네 개를 추가 사용하여 2×2 크기의 가방 두 개를 채우는 경우**에서, 1×1 크기의 물건 네 개는 이미 사용한 모든 1×1 크기의 물건이 1×1 크기의 가방에 들어가는 경우에만 강제적으로 사용해야 합니다.
- 그렇지 않은 경우, 즉 이미 1×1 크기의 가방의 개수보다 4개 이상 많은 1×1 크기의 물건을 사용한 경우, 이미 사용한 1×1 크기의 물건 네 개를 사용해도 좋습니다.

- 이 모든 경우를 생각하면 가능한 답의 모든 경우를 고려하게 됩니다.

E. 마비노기 가방 정리하기

- 물건에 대한 쿼리가 있는 경우를 생각합시다.
- 쿼리 처리 과정에서 필요한 연산은 다음과 같습니다.
  - 현재 가지고 있는 물건 중 가치가 k 이상인 가방 개수
  - 현재 가지고 있는 물건 중 가치가 높은 가방 k개의 가치의 합
  - 현재 가지고 있는 물건 중 가치가 k 번째로 높은 가방의 가치
- 먼저 쿼리에서 주어지는 모든 가방의 가치를 저장하여 정렬해 둔 뒤, 이를 이용해 구성한 세그먼트 트리를 이용합니다.
- 위의 연산을 모두 제공하는 세그먼트 트리를 구현할 수 있습니다.

- 시간 복잡도는 $\mathcal{O}\left(Q\log^2 N\right)$ 입니다.
- 그러나 상수가 크기 때문에, 구현에 따라 다소 최적화가 필요할 수 있습니다.
- 위의 연산은 모두 펜윅 트리에서 지원하기 때문에, 세그먼트 트리를 펜윅 트리로 교체할 수 있습니다.

F. 트리의 개수

`tree-dp`
출제 의도 – **Medium**

- 제출 12번, 정답 1팀 (정답률 8%)
- 처음 푼 팀: 가끔씩 툭하고 C언어로 부끄러워하는 옆자리의 이성호 양, 215분
- 출제자: 이동현

- $A_i \leq j$ 인 모든 정점 i 를 제거했을 때, 여러 트리로 나뉘게 됩니다.
- 남은 각 트리에 대해 독립적으로 생각할 수 있습니다. 그러므로 $A_i > j$ 인 모든 정점 i 가 하나의 트리를 구성한다고 생각합시다.
- 답에 더해지는 부분 트리의 개수를 구해야 합니다.

F. 트리의 개수

- 부분 트리가 $A_i > j$ 인 정점을 하나라도 포함하면 답에 더해집니다.
- 전체 부분 트리 개수에서 $A_i > j$ 인 정점을 하나도 포함하지 않는 부분 트리의 개수를 빼서 구합시다.
- 이를 위해 루트가 있는 트리에서 모든 정점에 대해 서브 트리의 부분 트리 개수를 동적 계획법으로 구해놓습니다.
- 모든 정점에 대해 서브 트리에 속한 정점을 제외한 트리의 부분 트리 개수도 구할 수 있습니다.

F. 트리의 개수

- 이제 각 간선 기준으로 답을 구합시다.
- 간선이 u와 v를 연결할 때, $A_u \leq A_v$라고 가정합시다.
- j가 A_u 이상 A_v 미만일 때, u 방향에 있는 정점들만 포함하는 부분 트리의 개수를 빼야합니다.
- 이는 사전에 구해놓은 값을 활용해 구할 수 있습니다.
- 시간 복잡도는 $\mathcal{O}(N \log N)$입니다.

G. 금고 털이 2

`ad-hoc, centroid, dp`

출제 의도 – **Hard**

- 제출 13번, 정답 1명 (정답률 8%)
- 처음 푼 팀 : 가끔씩 툭하고 C언어로 부끄러워하는 옆자리의 이성호 양, 180분
- 출제자 : 박영우

G. 금고 털이 2

- 루트의 차수가 2 이하이고 크기가 33인 트리 세 개를 만들고, 한 정점에 세 개의 트리의 루트를 붙이고 이 정점을 합쳐진 트리의 루트로 정합니다. 이렇게 하면 이 트리는 크기가 100이고 센트로이드는 루트가 됩니다.
- 만약 이 트리에서 간선을 제거하여 트리를 두 부분으로 나눌 경우, 나눠진 두 컴포넌트 중 큰 쪽은 반드시 센트로이드를 포함하는 쪽입니다.
- 간선을 제거한 후 남은 트리의 센트로이드는 원래 트리의 센트로이드가 됩니다.

G. 금고 털이 2

- 루트가 있는 이진트리 세 개를 만들어 루트를 하나의 정점에 붙인 크기 100의 트리를 만드는 풀이를 생각할 수 있습니다.
- 이렇게 할 경우 세 이진트리 중 최대 하나가 훼손되고, 두 개의 이진트리는 반드시 복원할 수 있습니다.
- 루트가 있는 크기 33의 이진트리의 개수는 약 3×10^9 개입니다.
- 주어진 비밀번호를 X 라 하고, $X = 10^9 A + B$ 꼴로 나타냅시다. 세 개의 수 $3A, 3B+1, 3(A xor B)+2$ 를 이진트리로 인코딩하여 전달하면 간선이 제거되어 셋 중 어느 하나의 수를 알아낼 수 없더라도 A, B 를 복원할 수 있습니다.
- A 와 B 는 10^9 이하의 정수가 될 수 있으므로 10^{18} 이하의 양의 정수를 전달할 수 있습니다.

H. 초전도체 부수기

ad-hoc
출제 의도 – **Easy**

- 제출 179번, 정답 8팀 (정답률 4%)
- 처음 푼 팀: 경곽 카와이 3D 아이돌 도전기, 34분
- 출제자: 채이환

- 최종적으로 K개의 조각의 무게는 하나만 $N - k + 1g$이며 나머지는 모두 $1g$인 최적해가 존재합니다.
- 쪼개는 과정을 역순으로 생각해보면 합치는 과정이 됩니다.
- 합치는 과정은 트리로 나타낼 수 있으며, 각 조각마다 트리에서 깊이에 해당하는 횟수만큼 비용에 무게가 더해집니다.
- 가장 깊이가 작은 조각이 가장 무겁도록 하는 것이 이득이므로 항상 그러한 최적해가 존재합니다.

H. 초전도체 부수기

- 하나만 무게가 $N-k+1g$ 이며 나머지는 모두 $1g$ 인 조각들을 합치는 비용의 최솟값을 구하면 됩니다.
- boj 13975 파일합치기 문제의 그리디를 적용할 수 있습니다.
- 무게가 같은 조각들이 많은 성질을 이용하여 set 등의 자료 구조를 통해 빠르게 답을 구할 수 있습니다.
- (무게, 개수) 쌍을 계속 관리하면 됩니다.

I. 순찰 업무

constructive, case-work
출제 의도 – **Hard**

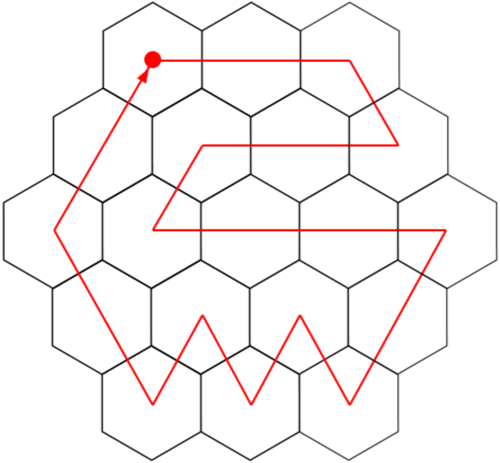
- 제출 11번, 정답 0팀 (정답률 0%)
- 처음 푼 팀: 없음
- 출제자: 문정후

Special Thanks to ...

- $K = 1$ 이고 시작점이 $(1, 1)$ 인 경우를 먼저 생각합니다.
- 여러 가지 방법이 있지만, 그중 하나의 방법은 다음과 같습니다.
- $(1, 1)$ 에서 시작합니다.
- 홀수 번째 행은 오른쪽으로, 짝수 번째 행은 왼쪽으로 채워 나갑니다. 이때 첫 번째 열은 채우지 않습니다.
- $b + c$ 의 값이 홀수인 경우 하나의 행이, 짝수인 경우 두 행이 마지막에 남게 됩니다.
  - 하나의 행이 남은 경우 그 행을 왼쪽으로 채웁니다.
  - 두 행이 남은 경우 지그재그 모양으로 두 행을 왼쪽으로 채웁니다.

- 마지막으로, 첫 번째 열을 위쪽으로 채워 $(1,1)$에 도착합니다.
- 이 경로를 P라고 하고, P_1에서 시작하여 $P_1, P_2, \cdots, P_M$을 순차로 지난다고 합시다.
- $P_1 = (1,1)$입니다.
- 편의상 k가 0 이하이거나 M 초과인 경우 P_k를 생각할 때 k에 적당한 K의 배수를 더하여 생각합니다.
- 이 경로는 아래 그림과 같습니다.

I. 순찰 업무



- 이를 통해, 시작점이 임의의 (x, y) 인 경우 역시 해결할 수 있습니다.
- 이미 시작점이 $(1, 1)$ 인 경우의 사이클을 만들었습니다.
- 이 경로는 모든 격자를 지나기 때문에, (x, y) 역시 지납니다.
- 그런데 이 경로는 사이클이므로, 시작점이 중요하지 않습니다.
- 즉, $P_k = (x, y)$ 인 k 가 존재합니다. $Q_1 = P_k, Q_2 = P_{k+1}, \cdots, Q_M = P_{k-1}$ 로 둡니다.
- Q 는 찾는 경로가 됩니다.

- 격자 X 를 시각 $Kn + t$(단, $0 \le t < K, t, n$ 은 정수)에 방문하는 것을 $X\langle t\rangle$ 로 나타냅니다.
- 즉, 모든 격자 X 에 대해 $X\langle 0\rangle, X\langle 1\rangle, \cdots, X\langle K-1\rangle$ 여야 합니다.

- K 가 홀수인 경우를 생각해 봅시다.
- 이미 위에서 (x, y) 에서 출발하는 경로 Q 를 얻었습니다.
- 다음을 $i = 1, 2, \cdots, M$ 에서 반복합니다.
- $Q_i, Q_{i+1}, Q_i, Q_{i+1}$, ..., Q_i 로 이동합니다. 이때 Q_i 가 $(k+1)/2$ 회, Q_{i+1} 가 $(k-1)/2$ 회입니다.
- 이것은 모든 짝수 p 에 대하여 $Q_i\langle p\rangle$, 그리고 모든 홀수 q 에 대하여 $Q_{i+1}\langle q\rangle$ 입니다.
- 즉, 모든 $i = 1, 2, \cdots, M$ 및 $j = 1, 2, \cdots, K$ 에 대하여 $R_{i(K-1)+j} = \begin{cases} Q_i (j\text{ 가 홀수}) \\ Q_{i+1} (j\text{ 가 짝수}) \end{cases}$ 로 둡니다.
- 모든 격자 X, $0 \le t < K$ 에 대해 $X\langle t\rangle$ 이므로 R 는 조건에 맞는 경로입니다.

- K 가 짝수, M 이 홀수인 경우를 생각해 봅시다.
- 이미 위에서 (x, y) 에서 출발하는 경로 Q 를 얻었습니다.
- 다음을 $i = 1, 2, \cdots, M$ 에서 반복합니다.
- $Q_{2i-1}, Q_{2i}, Q_{2i-1}, Q_{2i}, ..., Q_{2i-1}, Q_{2i}$ 로 이동합니다. 이때 Q_{2i-1} 가 $\frac{k}{2}$ 회, Q_{2i} 가 $\frac{k}{2}$ 회입니다.
- 이것은 모든 짝수 p 에 대하여 $Q_{2i-1}\langle p \rangle$, 그리고 모든 홀수 q 에 대하여 $Q_{2i}\langle q \rangle$ 입니다.
- 즉, 모든 $i = 1, 2, \cdots, M$ 및 $j = 1, 2, \cdots, K$ 에 대하여 $R_{i(K-1)+j} = \begin{cases} Q_{2i-1} (j \text{ 가 홀수}) \\ Q_{2i} (j \text{ 가 짝수}) \end{cases}$ 로 둡니다.
- 모든 격자 X, $0 \le t < K$ 에 대해 $X\langle t \rangle$ 이므로 R 는 조건에 맞는 경로입니다.

- K가 짝수, M이 짝수인 경우를 생각해 봅시다.
- 이미 위에서 (x, y)에서 출발하는 경로 Q를 얻었습니다.
- 이 경우에는 Q_k와 Q_{k+2}가 이웃하고, Q_{k-1}과 Q_{k+1}가 이웃하는 위치 k가 필요합니다. 이것은 위의 방법에서는, 경우에 따라 발생하지 않을 수 있습니다.
- $b + c$의 값이 짝수인 경우, 그러한 위치가 마지막 두 행에서 지그재그 모양으로 움직이는 경우에 발생합니다.
- a, b, c의 기우성을 모두 따져 봤을 때, $a + b, b + c, c + a$가 모두 홀수일 수는 없습니다. 이 중 짝수인 것이 존재해야 합니다.
- 그러한 값이 $b + c$가 되도록 주어진 육각형을 적당히 회전합니다. 이제 위치 k가 반드시 존재합니다.
- $Q'_1 = Q_k, Q'_2 = Q_{k+1}, \cdots, Q'_M = Q_{k-1}$로 둡니다.

- 다음을 $i = 1, 2, \cdots, \frac{M}{2}$ 까지만 반복합니다.
- 먼저, $Q'_{2i-1}, Q'_{2i}, Q'_{2i-1}, Q'_{2i}, \cdots, Q'_{2i-1}, Q'_{2i}$ 로 이동합니다. 이때 Q'_{2i-1} 가 $\frac{k}{2}$ 회, Q'_{2i} 가 $\frac{k}{2}$ 회입니다.
- 이것은 모든 짝수 p 에 대하여 $Q'_{2i-1}\langle p\rangle$, 그리고 모든 홀수 q 에 대하여 $Q'_{2i}\langle q\rangle$ 입니다.
- 그 후, $Q'_2, Q'_1, Q'_2, Q'_1, \cdots, Q'_2, Q'_1$ 으로 이동합니다. 이때 Q'_1 이 $\frac{k}{2}$ 회, Q'_0 이 $\frac{k}{2}$ 회입니다.
- 이것은 모든 짝수 p 에 대하여 $Q'_2\langle p\rangle$, 그리고 모든 홀수 q 에 대하여 $Q'_1\langle q\rangle$ 입니다.
- 이러한 이동이 가능한 것은 Q'_M 과 Q'_2 이 이웃하기 때문입니다.

- 마지막으로, 다음을 $i = \frac{M}{2}, \frac{M}{2} - 1, \cdots, 2$ 까지만 반복합니다.
- $Q'_{2i}, Q'_{2i-1}, Q'_{2i}, Q'_{2i-1}, \cdots, Q'_{2i}, Q'_{2i-1}$ 로 이동합니다. 이때 Q'_{2i} 가 $\frac{k}{2}$ 회, Q'_{2i-1} 가 $\frac{k}{2}$ 회입니다.
- 이것은 모든 짝수 p 에 대하여 $Q'_{2i}\langle p\rangle$, 그리고 모든 홀수 q 에 대하여 $Q'_{2i-1}\langle q\rangle$ 입니다.
- Q'_3 과 Q'_1 이 이웃하므로, 이 경로 역시 사이클이 됩니다.
- 위의 방법과 같은 방법으로, (x, y) 에서 시작하도록 하는 경로 R 를 만듭니다.
- 모든 격자 X, $0 \le t < K$ 에 대해 $X\langle t\rangle$ 이므로 R 는 조건에 맞는 경로입니다.

J. 줌

`data-structure, segtree, lazyprop`

출제 의도 – **Challenging**

- 제출 9번, 정답 0명 (정답률 0%)
- 처음 푼 팀: 없음
- 출제자: 박영우

3번 쿼리가 없을 때의 풀이

- 1번 쿼리는 배열의 $[l, r]$ 원소에 특정한 수 b를 더하는 쿼리인데, 이는 $[1, l-1]$ 번째 원소에 b를 빼고, $[1, r]$ 번째 원소에 b를 더하는 것으로 생각할 수 있습니다.
- 따라서 배열을 $(ind_1, b_1), (ind_2, b_2), \cdots (ind_k, b_k)$ 순서쌍들의 합으로 표현합시다. 이 경우, 모든 원소가 0인 배열에 각 순서쌍 (ind_i, b_i) 에 대하여 첫 l_i 개의 원소에 b_i 를 더한 배열을 뜻합니다.
- 순서쌍 (ind_i, b_i) 은 2번 쿼리로 인해 $(l + \lfloor\{\frac{ind_i - 1}{2}\}\rfloor)$ 로 변합니다.

J. 줌

- 순서쌍들의 집합을 관리하고, 순서쌍들이 1번 쿼리로 추가되거나 2번 쿼리로 변경될 때마다 배열을 세그먼트 트리로 관리하는 풀이를 생각할 수 있습니다.
- 이때 2번 쿼리로 인하여 순서쌍의 ind값이 인덱스를 벗어난 경우는 집합에서 제거하고, 2번 쿼리로 인해 두 순서쌍의 ind값이 합쳐진 경우 두 순서쌍을 합해 줍시다.
- 세 순서쌍 $(ind_1, b_1), (ind_2, b_2), (ind_3, b_3)$에 대하여, $\log_2 N$번의 2번 쿼리를 거쳤을 때 세 순서쌍 중 적어도 하나가 제거되거나 다른 순서쌍과 합쳐진다는 사실을 알 수 있습니다.
- 따라서 구간을 변경하는 횟수는 총 $O(Q \log_2 N)$회이고, 세그먼트 트리를 사용하므로 전체 시간복잡도는 $O(Q \log_2^2 N)$입니다.

J. 줌

3번 쿼리가 있을 때의 풀이

- 길이 N의 두 배열 A와 B에 대하여 두 배열을 원소별로 더한 배열을 $A+B$로 정의합시다. 또한 모든 원소가 0인 배열을 O라 정의합시다.
- 길이 N의 배열 A에서 현재까지 등장한 모든 1, 2번 쿼리를 실행한 배열을 $T(A)$, 2번 쿼리만을 실행한 배열을 $T_2(A)$라 합시다.
- 만약 길이 N의 배열에서 3번 연산을 실행한다면, 배열은 $T_2(A)+T(O)=T_2(A)+A$가 된다는 사실을 관찰할 수 있습니다.
- 즉, 3번 쿼리가 실행된다면 배열 A에 A에서 2번 쿼리를 다시 실행한 배열이 더해진다는 사실을 알 수 있습니다.

J. 줌

- 현재까지 실행한 2번 쿼리가 t회일 때, 어떤 배열 A에 대하여 $T_2(A)$는 최대 $\max\{2, N/2^t\}$ 개의 원소로 이루어진다는 사실을 알 수 있습니다.
- 따라서 2, 3번 쿼리가 주어질 때마다 T_2를 갱신하고 3번 쿼리가 주어질 때마다 $T_2(A)$를 여러 개의 (ind_i, b_i) 형태의 순서쌍으로 바꿔 더하는 방식으로 풀 수 있습니다.
- 첫 $O(\log_2 N)$회의 3번 쿼리로 인해 총 $O(N)$개의 순서쌍이 추가되고, 이후의 3번 쿼리들로 인해 $O(Q)$개의 순서쌍이 추가됩니다.
- 따라서 전체 시간 복잡도는 $O((N+Q)\log_2^2 N)$입니다.
- 첫 2번 쿼리 전의 3번 쿼리는 $T_2(A)$를 이루는 원소의 가짓수에 영향을 주지 않으므로 이 경우를 예외처리해야 합니다.

K. 카드 색칠 2

`math, case-work`

출제 의도 – **Medium**

- 제출 13번, 정답 2명 (정답률 15%)
- 처음 푼 팀: 강상협 조은호, 125분
- 출제자: 박영우

K. 카드 색칠 2

- 각 행을 이루는 흰색 카드의 개수가 같으므로, i 행 n 열의 카드는 $i+1$ 행 1 열의 카드와 같다는 것을 알 수 있습니다.
- 따라서 $i+1$ 행의 카드 배열은 i 행의 마지막 카드를 맨 앞으로 옮긴 것과 같습니다.
- 따라서 첫 행의 카드 색깔이 정해지면 전체 카드들을 색칠하는 방법은 유일합니다.
- 첫 행이 모두 흰색이라면 전체 카드는 하나의 문양을 형성합니다. 그렇지 않은 경우만 고려합시다.

K. 카드 색칠 2

- 첫 행만 고려할 때 생기는 연속한 흰색 카드 구간이 전체 카드 배열에서 총 몇 개의 문양을 만드는지 생각해봅시다.
- 하나의 카드로만 이루어진 구간의 경우, 그 카드 하나만이 문양을 이룹니다. 다른 행에서도 마찬가지입니다.
- 따라서 총 N 개의 문양이 생깁니다.

K. 카드 색칠 2

- 하나 이상의 카드로만 이루어진 구간의 경우, 문양은 총 2개가 생깁니다.
- 예외가 있습니다. 첫 몇 개의 카드가 흰색이고 마지막 몇 개의 카드가 흰색인 경우, 첫 몇 개의 카드들로 인해 생기는 문양들과 마지막 몇 개의 카드들로 인해 생기는 문양들이 합쳐집니다.
- 이 경우, 총 두 개의 구간을 합하여 세 개의 문양이 형성됩니다.

K. 카드 색칠 2

- 위 사실들을 이용하면 첫 행의 카드들의 색깔이 확정되었을 때 $O(N)$의 시간에 답을 구할 수 있습니다.
- 더블 카운팅을 사용하면 전체 문제를 $O(N)$에 처리할 수 있습니다.
- 각 카드가 크기 1인 흰색 구간을 형성하는 경우, 크기 1 이상인 흰색 구간의 끝점을 형성하는 경우 전체 카드를 색칠하는 방법의 가짓수를 구하면 됩니다.
- 전체 카드가 흰색인 경우, 첫 번째 카드들과 마지막 카드들이 모두 흰색이 될 경우의 예외 처리를 주의해야 합니다.

L. 정기 모임 5

ad-hoc
출제 의도 – **Easy**

- 제출 12번, 정답 1팀 (정답률 8%)
- 처음 푼 팀: 경곽 카와이 3D 아이돌 도전기, 223분
- 출제자: 이동현

- 트리의 지름의 길이를 L 이라고 합시다.
- 트리의 지름에 있는 정점들을 생각해보면 k 가 $\lfloor L/2 \rfloor$ 이상인 것을 알 수 있습니다.
- 처음 트리에서 지름의 중심에 있는 정점 중 하나를 M 번 정점이라고 합시다. M 번 사람이 사는 정점에 사는 사람을 선택하는 것을 반복하는 길이 $\lfloor L/2 \rfloor$ 의 p 가 항상 존재합니다.

L. 정기 모임 5

- M 번 사람이 사는 정점에 사는 사람 중 아직 선택하지 않은 사람들의 번호를 관리하는 자료 구조를 사용합시다.
- 자료 구조가 저장하고 있는 번호 중 최댓값 또는 최솟값을 출력하고, 새로운 번호들을 추가하는 과정을 반복합니다.
- 최댓값을 먼저 출력할지, 최솟값을 먼저 출력할지는 두 번째 출력에서 결정하면 됩니다.
- 시간 복잡도는 $\mathcal{O}(N \log N)$ 입니다.