

가희와 함께 하는 6회 코딩테스트

문제	의도한 난이도	출제자
A gahui and sousenkyo 1	Easy	chogahui05
B gahui and sousenkyo 2	Easy	chogahui05
C gahui and sousenkyo 3	Easy	chogahui05
D 가희와 클럽 오디션 1	Easy	chogahui05
E gahui and sousenkyo 4	Easy	chogahui05
F 가희와 클럽 오디션 2	Medium	chogahui05
G 가희와 여행가요	Medium	chogahui05
H 가희와 터널	Hard	chogahui05
I 가희와 총선거 5	Hard	chogahui05
J gahui and sousenkyo 6	Challenging	chogahui05
K 가희와 집	Challenging	chogahui05

A. gahui and sousenkyo 1

implementation

출제진 의도 – **Easy**

- ✓ 제출 146, 정답 125명 (정답률 86.301%)
- ✓ 처음 푼 사람: **kiwiyou**, 1분
- ✓ 출제자: chogahui05

A. gahui and sousenkyo 1

- ✓ Tier 1에 속하지 않는다는 의미는, 상위 16위에 속하지 않는다는 것입니다.
 - 1번째 수는 16위의 득표수를
 - 2~5번째 수는 17~20위의 득표수를 의미합니다.
- ✓ 2~5번째 수 중, 1번째 수에서 1000을 뺀 값보다 크거나 같은 수가 몇 개인지 세면 됩니다.

B. gahui and sousenkyo 2

implementation, sort

출제진 의도 – **Easy**

- ✓ 제출 166번, 정답 120명 (정답률 73.494%)
- ✓ 처음 푼 사람: **kiwiyou**, 3분
- ✓ 출제자: chogahui05

B. gahui and sousenkyo 2

- ✓ 캐릭터의 순위를 구해야 합니다.
- ✓ 득표수가 높으면 랭킹이 높은 것입니다. 따라서
 - 내가 응원하는 캐릭터의 득표수를 v 라 하면 득표수가 v 보다 많은 캐릭터 수를 구합니다.
 - 이 값에 1을 더하면 됩니다.
- ✓ 물론, $\mathcal{O}(n^2)$ 정렬을 사용해도 됩니다.

C. gahui and sousenkyo 3

implementation

출제진 의도 – **Easy**

- ✓ 제출 115 번, 정답 106명 (정답률 93.043%)
- ✓ 처음 푼 사람: **kiwiyou**, 6분
- ✓ 출제자: chogahui05

C. gahui and sousenkyo 3

- ✓ 캐릭터 유형이 몇 개 있습니다. 이 유형이 무엇에 따라 결정되나요?
 - 속보에서 획득한 표수를 p_x
 - 본선에서 획득한 표수를 r_x
 - p_x 를 r_x 로 나눈 값에 따라 결정됩니다.
- ✓ 그러면 이 값을 구한 다음에, 조건에 맞게 처리하면 됩니다.

C. gahui and sousenkyo 3

- ✓ 실수 오차 때문에 틀리는 것이 걱정되신다면 어떻게 하면 될까요?
- ✓ 예를 들어 p_x 를 r_x 로 나눈 값이 0.2 보다 작은지 검사해야 한다고 합시다.
 - $\frac{p_x}{r_x} < 0.2$ 에서 양 변에 100을 곱하면 $\frac{100 \cdot p_x}{r_x} < 20$ 이 됩니다.
 - 그러면, $100 \cdot p_x$ 를 r_x 로 나눈 몫이 20 보다 작은지 보면 되겠네요?
- ✓ 가희와 방어울 무시 문제를 연계했으니, 같이 풀어보시면 좋습니다.

D. 가희와 클럽 오디션 1

implementation

출제진 의도 – **Easy**

- ✓ 제출 78, 정답 68명 (정답률 88.462%)
- ✓ 처음 푼 사람: **tony9402**, 5분
- ✓ 출제자: chogahui05

D. 가희와 클럽 오디션 1

- ✓ 5가지 판정이 있습니다.
- ✓ 이 중 *perfect* 판정은 n 연팩 상태가 있습니다.
 - n 이 2 이상이라면 이전에도 *perfect*를 받아야 합니다.
 - 그런데, 이전 판정과 다르다고 했습니다. 따라서, n 은 2 이상일 수 없습니다.
- ✓ 고로, 각 판정마다 점수에 키 노트의 lv 을 곱하면 됩니다.

E. gahui and sousenkyo 4

greedy

출제진 의도 – **Easy**

- ✓ 제출 74 번, 정답 31명 (정답률 44.595%)
- ✓ 처음 푼 사람: **dabbler1**, 12분
- ✓ 출제자: chogahui05

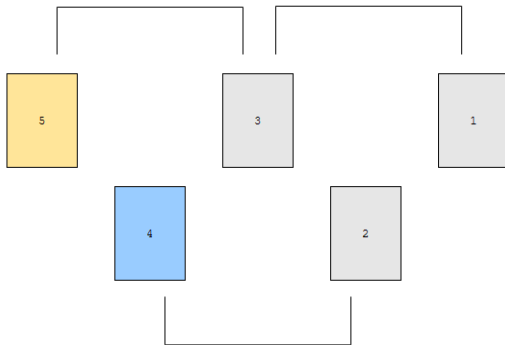
E. gahui and sousenkyo 4

- ✓ 같은 표수를 득한 캐릭터가 없습니다.
- ✓ 표수 차이가 k 인 두 캐릭터는 없습니다.
- ✓ 쉬워 보이지는 않습니다만. $k = 1$ 인 경우부터 접근해 봅시다.

E. gahui and sousenkyo 4

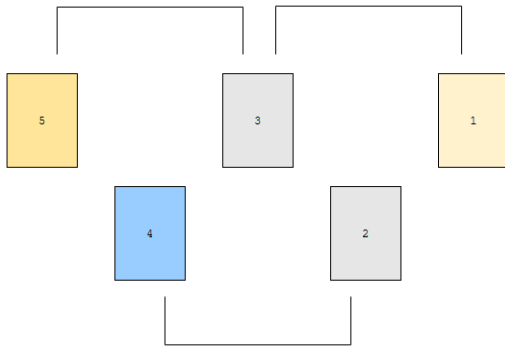
- ✓ $k = 2$ 이고, $v = 5$ 라고 하겠습니다.
 - 1위는 5표를 획득했습니다.
 - 2위는 4표를 획득할 수 있나요?
 - ▶ 1위가 5표를 획득했던, 그렇지 않았던 2위는 4표를 획득할 수 있습니다.
 - ▶ 그러면, 여기서 5와 4는 독립적으로 작용하고 있음을 볼 수 있습니다.
 - 이걸 그림으로 그려보겠습니다.

E. gahui and sousenkyo 4



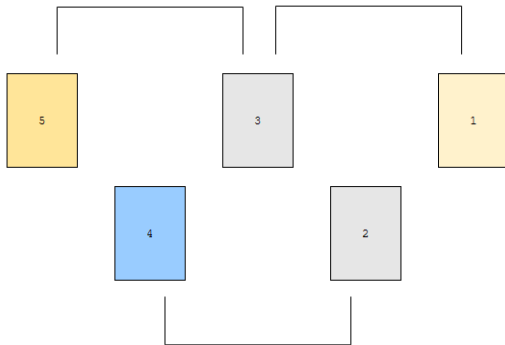
- ✓ 이것을 잘 보면 $5 - 3 - 1$ 묶음과 $4 - 2$ 묶음이 독립적이라는 것을 확인할 수 있어요.
- ✓ 위 그림은 1위가 5표, 2위가 4표를 획득한 상황을 그린 거예요.

E. gahui and sousenkyo 4



- ✓ 3위는 3표를 획득 못해요. 왜?
- ✓ 이미 5표를 획득한 캐릭터가 있기 때문이에요. 대신에 1표는 획득 가능합니다.

E. gahui and sousenkyo 4



✓ 즉, 독립적인 요소에 대해 홀 - 짝 - 홀 - 짝이 적용되는 케이스입니다.

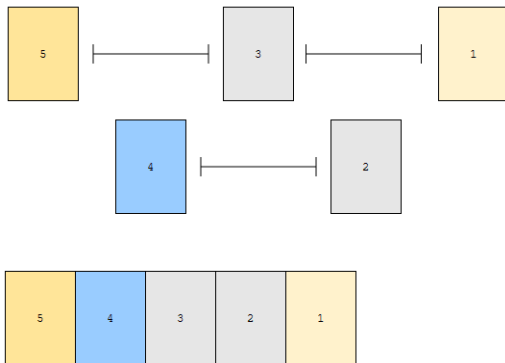
E. gahui and sousenkyo 4

- ✓ 이제 이 문제를 풀어봅시다.
 - 1 부터 v 까지 v 개의 수가 있습니다.
 - 수 v 를 뽑았다면, 1 차이나는 수를 뽑지 못해요.
 - 뽑은 수의 개수를 최대화 하고 싶어요.

E. gahui and sousenkyo 4

- ✓ 어떻게 풀까요? 간단합니다.
 - v 를 뽑습니다. 왜?
 - ▶ v 를 버리고 $v - 1$ 를 뽑으면
 - ▶ v 를 버렸기 때문에, 기회가 하나 날라가게 됩니다.
 - v 를 뽑았다면, $v - 1$ 은 못 뽑습니다.
 - $v - 2$ 는 뽑을 수 있습니다. $v - 2$ 를 뽑아요. 왜? 비슷한 논리로 생각하시면 됩니다.

E. gahui and sousenkyo 4



✓ 이제 구현해 봅시다.

E. gahui and sousenkyo 4

- ✓ 이전 페이지에서 규칙을 찾으셨겠지만
 - 5표의 경우 7표를 획득한 캐릭터가 없어서 넣을 수 있습니다.
 - 4표의 경우 6표를 획득한 캐릭터가 없어서 넣을 수 있습니다.
 - 그런데 3표의 경우 5표를 획득한 캐릭터가 있어서 넣을 수 없습니다.
 - 2표의 경우도 마찬가지겠군요.
 - 1표의 경우 3표를 획득한 캐릭터가 없으니 넣으면 되겠네요.

E. gahui and sousenkyo 4

- ✓ 고로, 최종 로직은 아래와 같습니다.
 - v 표를 획득했다고 표시합니다.
 - $v - 1$ 표부터 1 표에 대해 아래를 반복합니다.
 - ▶ 검사하고 있는 표 수가 x 라면
 - ▶ $x + k$ 표를 획득한 캐릭터가 있으면 건너 뛩니다.
 - ▶ 그렇지 않으면 x 표를 획득했다고 표시합니다.

F. 가희와 클럽 오디션 2

implementation parsing

출제진 의도 – **Medium**

- ✓ 제출 55번, 정답 19명 (정답률 36.364%)
- ✓ 처음 푼 사람: **dabbler1**, 24분
- ✓ 출제자: chogahui05

F. 가희와 클럽 오디션 2

- ✓ 카카오 기출 문제인 **방금 그 곡**과 비슷한 문제입니다.
- ✓ 파싱 방법은 **방금 그 곡**과 같습니다.
- ✓ 일일이 파싱하거나, *replace*로 특정한 문자로 바꾸거나

F. 가희와 클럽 오디션 2

- ✓ *replace*를 하는 경우 주의해야 할 점이 있습니다.
 - *LU*를 *LU!*보다 먼저 *replace*하면 안 됩니다. 왜?
 - *LU*를 7로 *replace*했다고 합시다.
 - 7!은 무엇을 의미하나요?
- ✓ 고로, 길이가 긴 것부터 순차적으로 *replace* 시킵니다.

F. 가희와 클럽 오디션 2

- ✓ 이 *replace* 일일이 쓰기 귀찮습니다.
- ✓ 이런 것은 *ordered_map* 등으로 쉽게 관리할 수 있습니다.
- ✓ *iteration* 을 돌면서 *replace* 를 시키면 깔끔하게 코딩 가능합니다.
- ✓ 이제 남은 것은 상태 처리입니다.

F. 가희와 클럽 오디션 2

- ✓ 여기서 빠질 수 있는 함정이 하나 있습니다.
 - *ends_with* 등으로 *suffix* 만 검사하면 될까요?
 - 예제 2는 그러면 안 된다는 것을 보여줍니다.
- ✓ 그러면 어떻게 처리해야 하는가? 일일이 키 노트와 내가 친 키를 비교해야 합니다.

F. 가희와 클럽 오디션 2

- ✓ 먼저 0개 이상 lv 개 미만의 키와 일치할 때.
 - i 개의 키와 일치할 때, $i + 1$ 번째 키와 내가 친 키가 일치하면 일치하는 키가 하나 증가합니다.
 - 그렇지 않으면 0개가 일치합니다.
- ✓ lv 개의 키가 일치하면 어떤 키를 눌러도 0개가 되는 것만 조심하면 됩니다.

G. 가희와 여행가요

mst, graph theory, mst, union find

출제진 의도 - **Medium**

- ✓ 제출 53번, 정답 18명 (정답률 35.849%)
- ✓ 처음 푼 사람: **kiwiyou**, 37분
- ✓ 출제자: chogahui05

G. 가희와 여행가요

- ✓ $n - 1$ 개의 도시와 연합을 한다는 것은 무엇을 의미할까요?
- ✓ 월드 내에 있는 도시들과 가희의 도시가 한 집합에 있다는 것입니다.
- ✓ 그러면 쓸 수 있는 것은 *union_find* 인데요. 조금 더 문제를 읽어 봅시다.

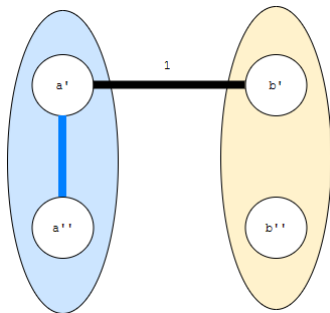
G. 가희와 여행가요

- ✓ 총 건설 비용이 최소가 되게 하고 싶어요. mst 네요.
- ✓ 그런데 최대한 빠르게 목적을 달성하고 싶어요.
 - t 이전에 건설할 수 있는 노선으로 mst 를 구축해 봅시다. 총 비용을 c_t 라 합시다.
 - ▶ x 이전에 건설할 수 있는 노선들의 집합을 S_x
 - ▶ y 이전에 건설할 수 있는 노선들의 집합을 S_y 라 할게요.
 - ▶ $x < y$ 이면 $S_x \subseteq S_y$ 입니다.
 - ▶ S_y 에서 노선을 선택하는 가짓수는, S_x 에서 선택할 수 있는 가짓수를 포함합니다.
 - ▶ 고로, $c_y \leq c_x$ 입니다.
 - 단조 증가, 혹은 감소 함수이므로 이분 탐색을 적용할 수 있어요.

G. 가희와 여행가요

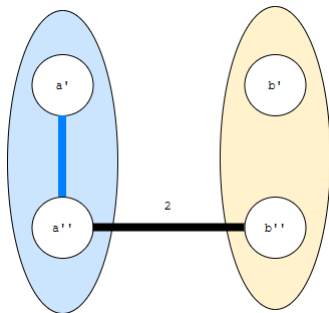
- ✓ 더 쉬운 방법이 있을까요?
- ✓ *kruskal* 알고리즘에서, *cost*가 최소인 것부터 뽑았어요. 이 때
 - 뽑힌 간선은 *mst*에 들어갑니다.
 - 추가로 요구하는 상황은 *time*을 최소화 시키라는 것입니다.
 - 이 경우, *cost*가 같으면 *time*이 작은 것부터 먼저 뽑게 하면 됩니다. 왜 그럴까요?

G. 가희와 여행가요



- ✓ 시간이 1 이고 cost가 c 인 간선을 선택한 상황입니다.
- ✓ 이 경우, 걸리는 시간은 1과 파란색 부분, 노란색 부분의 시간 max 값이 됩니다.

G. 가희와 여행가요



- ✓ 시간이 2 이고 cost가 c 인 간선을 선택한 상황입니다.
- ✓ 이 경우, 걸리는 시간은 2과 파란색 부분, 노란색 부분의 시간 max 값이 됩니다.
- ✓ 이전 상황보다 이득이 되지 않습니다.

G. 가희와 여행가요

- ✓ 그러므로 아래와 같은 로직을 구현하시면 됩니다.
 - *cost* 오름차순으로, *cost*가 같으면 *time* 오름차순으로 정렬합니다.
 - *union_find* 알고리즘을 이용해서 크루스칼 알고리즘을 적용합니다.
 - 혹은 *prim* 알고리즘을 적용하셔도 됩니다.
- ✓ 4분컷은 상상도 못했습니다.

G. 가희와 터널

dynamic_programming

출제진 의도 - **Hard**

- ✓ 제출 12번, 정답 11명 (정답률 91.667%)
- ✓ 처음 푼 사람: **p_ce1052**, 6분
- ✓ 출제자: chogahui05

H. 가희와 터널

- ✓ 가희가 빨-주-노-초-파-남-보 순서대로 먹게 하는 방법이 하나 이상 있다.
- ✓ 이 상황을 역으로 뒤집어 봅시다.
 - 가희가 빨-주-노-초-파-남-보
 - 순서대로 먹게 하는 방법이 아예 없다.
- ✓ 이 두 상황은 서로 **여사건** 관계입니다. 이를 이용해 봅시다.

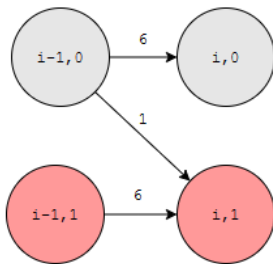
H. 가희와 터널

- ✓ $dp[lo][v]$ 를 아래와 같이 정의합니다.
 - 빨간색을 1 번, ... , 보라색을 7 번 간식이라 하겠습니다.
 - lo 번 위치까지 가희가 방문했을 때
 - 특별한 순서에 맞게 v 번 간식까지 먹게 할 수 있는 가짓수
 - 예외로 0은 아무 간식도 먹지 못한 가짓수로 정의합니다.

H. 가희와 터널

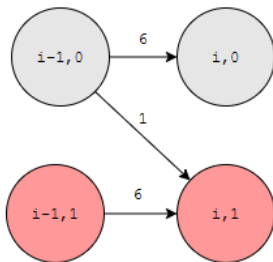
- ✓ 먼저, 0번 위치를 시작 위치라고 합시다.
 - 이 경우, 가희가 먹을 수 있는 간식이 아무것도 없습니다.
 - 0번 위치까지 방문했을 때
 - 어떻게 해도 간식을 먹지 못하니, $dp[0][0] = 1$ 로 정의합니다.

H. 가희와 터널



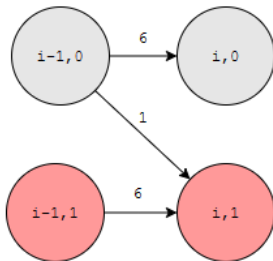
- ✓ 먼저, $i - 1$ 번 위치까지 가희가 빨간색을 못 먹었다고 해 보겠습니다.
 - i 번 위치에 빨간색이 놓이면 가희는 빨간색을 먹을 수 있어요. 1 가지군요.
 - 그렇지 않으면 여전히 빨간색을 못 먹은 상태겠지요? 이 경우가 6 가지입니다.

H. 가희와 터널



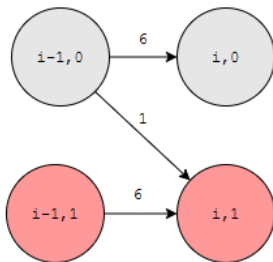
- ✓ 그러면 $dp[i][1]$ 를 어떻게 구하면 될까요?
 - i 번 위치에 빨간색이 놓이면 가희는 $i - 1$ 번까지 아무것도 못 먹은 상태에서
 - 빨간색을 먹을 수 있어요. 1 가지군요.
 - 고로 $dp[i][1]$ 은 $dp[i - 1][0]$ 에다가 무언가를 더한 상태라 볼 수 있어요.

H. 가희와 터널



- ✓ 그러면 $dp[i][1]$ 를 어떻게 구하면 될까요?
 - i 번 위치에 빨간색이 놓이면 가희는 $i - 1$ 번까지 아무것도 못 먹은 상태에서
 - 빨간색을 먹을 수 있어요. 1 가지군요.
 - 고로 $dp[i][1]$ 은 $dp[i - 1][0]$ 에다가 무언가를 더한 값이라 볼 수 있어요.

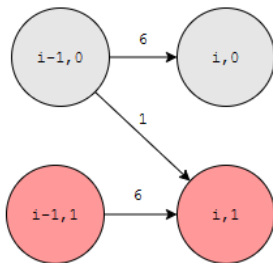
H. 가희와 터널



✓ 고려해야 하는 상태가 더 없나요?

- 가희가 $i - 1$ 번까지 빨간색만 먹은 상태도 있어요.
- 이 경우, i 번째에 주황색이 아닌 다른 것이 있으면 빨간색만 먹은 상태가 됩니다.
- 고로 $dp[i][1]$ 에 $6 \cdot dp[i - 1][1]$ 을 더해야 합니다.

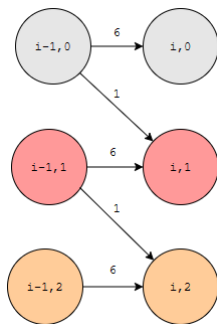
H. 가희와 터널



✓ 따라서 $dp[i][1]$ 은 다음 두 값을 더한 값입니다.

- $dp[i-1][0]$
- $6 \cdot dp[i-1][1]$

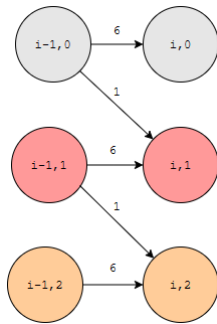
H. 가희와 터널



✓ $dp[i][2]$ 는 어떻게 구할까요?

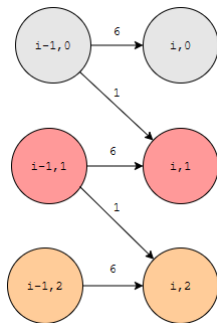
- i 번째까지 빨간색, 주황색 순으로 먹을 수 있게 하는 가짓수입니다.
- 이것도 경우가 2가지 있습니다. 어떤 경우?

H. 가희와 터널



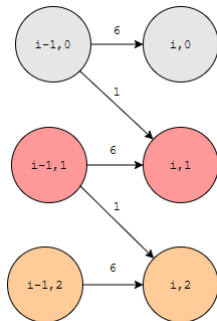
- ✓ $i - 1$ 번까지 빨간색만 먹게 할 수 있게 하는 경우 $dp[i - 1][1]$
- ✓ $i - 1$ 번까지 빨간색-주황색 순서대로 먹게 할 수 있는 경우 $dp[i - 1][2]$

H. 가희와 터널



- ✓ 전자의 경우 주황색이 오게 하면 됩니다.
- ✓ 후자의 경우 주황색 말고 다른 색의 간식이 오게 하면 됩니다.

H. 가희와 터널



✓ 따라서, $dp[i][2]$ 는 아래 두 값을 합산하면 됩니다.

- $dp[i-1][1]$
- $6 \cdot dp[i-1][2]$

H. 가희와 터널

- ✓ $dp[i][v]$ 는 같은 방법으로 생각하면 아래와 같이 구하면 됨을 알 수 있습니다.
 - $dp[i-1][v-1] + 6 \cdot dp[i-1][v]$
 - 예외로 v 가 0인 경우가 있습니다.
 - ▶ 이 경우, i 번 위치까지 빨간색 간식이 없는 경우와 같습니다.
 - ▶ 6^i 를 계산하면 됩니다.

I. 가희와 총선거 5

implementation, parsing

출제진 의도 - **Hard**

- ✓ 제출 27번, 정답 7명 (정답률 25.926%)
- ✓ 처음 푼 사람: **kiwiyou**, 71분
- ✓ 출제자: chogahui05

I. 가희와 총선거 5

- ✓ 사실, 사전 지식이 있었다면 매우 쉽게 풀었을 수도 있었던 문제였습니다.
- ✓ 독해해야 하는 지문이 꽤 줄어들기 때문입니다.
- ✓ 어디까지나 문제에 함정이 없었다는 전제입니다.

I. 가희와 총선거 5

- ✓ 만약에 *center* 의 정의를 제가 바꿨다면 어떻게 되었을까요?
 - 티어에서 제일 높은 순위가 아니라
 - 티어 구간에서의 순위 중앙값이 *center* 다라고 했다면?
 - 순위 구간이 뒤죽박죽 했다면?
- ✓ 사전 지식으로 밀면 안 되는 이유입니다.

I. 가희와 총선거 5

- ✓ 이 문제는 친절하게도 그렇지 않으니, 티어 구간은 사전 지식이 있었다면 스윙 하셨겠네요.
- ✓ 그렇지 않아도 금방 보실 수 있는데요.
 - 16위 단위로 끊기고 있어요.
 - 캐릭터의 랭킹을 r 이라 했을 때 $r - 1$ 이 16으로 떨어지면 센터네요.
 - 그리고 $r - 1$ 을 16으로 나누었을 때 몫이 티어랑 관련이 있네요. 여기까지 체크.

I. 가희와 총선거 5

- ✓ 최종 결과가 티어 6이라는 의미는 결과에 안 나타났단 의미입니다.
- ✓ 고로 신데렐라 수치를 구할 때에는 최종 결과에 나온 캐릭터만 구하면 됩니다.
- ✓ $x > y$ 번 조건부터 보겠습니다.
 - x 가 y 보다 크다면, 티어가 더 높아졌다는 것을 의미합니다.
 - 티어 1부터 6까지 피라미드를 그려보면 어떤 말인지 아실 수 있을 겁니다.

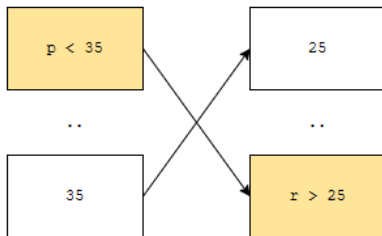
I. 가희와 총선거 5

- ✓ 이 조건과 물리는 것이 센터일 때인데요.
 - x 가 y 보다 크다면, 20 000이 더해집니다.
 - 그렇지 않으면 10 000이 더해지는군요. 여기까지 체크.
- ✓ 최종 티어가 1이 아닌 경우 여기서 끝내면 됩니다.

I. 가희와 총선거 5

- ✓ 티어 1인 경우 더 붙는 것이 있습니다. 이게 핵심입니다.
 - 예상 결과가 c 보다 좋았지만, 최종 결과가 c 보다 좋지 않다.
 - 예상 결과가 티어 1에 속했지만, 최종 결과가 티어 1에 속하지 않는다.
- ✓ 이 두 조건. 무슨 말을 하고 싶었던 것일까요?

I. 가희와 총선거 5



✓ 먼저 1번 조건입니다.

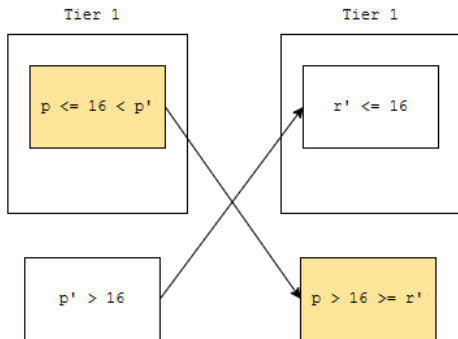
- c 는 노란색이고 하얀색은 35위에서 25위로 뛴 캐릭터입니다.
- 이 그림의 형태. x 자를 잘 기억해 두세요.

I. 가희와 총선거 5

✓ 2번 조건입니다.

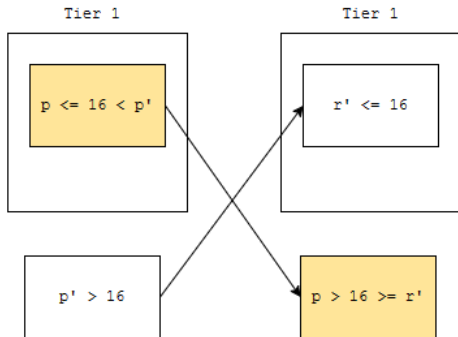
- 노란색은 예상이 티어 1 이었는데, 최종이 티어 1 이 아니었습니다.
- 그런데, $x > 1$ 이지만 $y = 1$ 이란 것은
 - ▶ 대상 캐릭터가 티어 1로 예상이 안 되었는데
 - ▶ 최종이 티어 1로 올라왔다는 것입니다.
- 그림은 어떻게 그려질까요?

I. 가희와 총선거 5



- ✓ 위와 같이 그려집니다. x 자가 그대로 나왔습니다.
- ✓ 당연하게도 1 번 조건을 포함합니다.

I. 가희와 총선거 5



- ✓ 그러면 p 는 어떻게 구하면 될까요?
- ✓ 예상 결과가 티어 1 이면서, 최종 결과는 티어 1 이 아닌 수를 구하면 됩니다.

I. 가희와 총선거 5

- ✓ 이제 입력이 어떻게 주어지는지 봅시다.
 - 캐릭터 정보가 공백으로 구분되어 주어지고
 - *name_info*와 *group_info*와 *team_info*가 공백으로 구분되는데
 - *name_info*가 공백이 나옵니다.
- ✓ *token* 개수가 *dynamic* 합니다.
- ✓ 이런 문제. 제가 몇 번 출제한 적이 있으니 처리 방법을 알아두시면 좋습니다.

I. 가희와 총선거 5

- ✓ 어떤 것이 얻을 수 있는 고정된 정보인가?
 - *group_info*와 *team_info*는 *token* 개수가 2로 동일해요.
 - 그러면 이것들은 어떻게 찾나요?
 - *Group*이 나오면 *group_info*가 시작되었다는 것을 알 수 있어요.
- ✓ 따라서, 아래와 같이 *parsing* 처리를 하시면 됩니다.

I. 가희와 총선거 5

- ✓ 토큰을 계속 받다가
 - 토큰 *Group*이 나오면 4개의 토큰을 받습니다.
 - 2번째 토큰은 *group_name*이 되고
 - 4번째 토큰은 *team_name*이 되겠네요.
- ✓ 계속 위 과정을 반복해서 받으면 됩니다.

I. 가희와 총선거 5

- ✓ 남은 것은 데이터 처리입니다.
 - 이름, 그룹, 팀이 모두 같은 경우는 없다고 되어 있습니다.
 - 고로 *name#group#team* 등으로 *key*를 만들어서 정보를 관리하면 됩니다.
- ✓ 같은 이름인 경우를 주의해서 처리하면 되겠습니다.

J. gahui and sousenkyo 6

constructive, math

출제진 의도 – **Challenge**

- ✓ 제출 40번, 정답 8명 (정답률 25.000%)
- ✓ 처음 푼 사람: **p_ce1052**, 26분
- ✓ 출제자: chogahui05

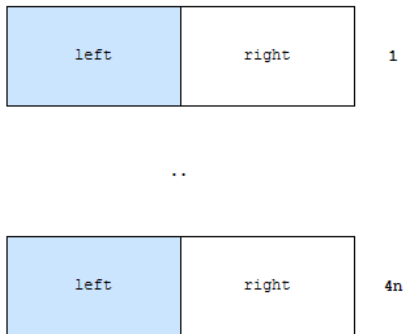
J. gahui and sousenkyo 6

- ✓ n 개의 수는 모두 다릅니다.
- ✓ n 개의 수를 *bitwise_xor* 했을 때 값은 x 입니다.
- ✓ 모두 다르다는 조건 때문에 쉽지 않은 문제입니다. 몇 가지 관찰을 해 봅시다.

J. gahui and sousenkyo 6

- ✓ 1부터 $4 \cdot k$ 까지 *bitwise_xor* 했을 때 값은 0입니다.
- ✓ 5부터 $4 \cdot k'$ 까지 *bitwise_xor* 했을 때 값도 0입니다.
- ✓ 이 말은 $4 \cdot n$ 까지 5부터 $4 \cdot k'$ 까지 순서대로 넣어도 된다는 의미입니다.
- ✓ *bit*를 왼쪽으로 이동시켜도 그 결과는 바뀌지 않습니다.
- ✓ 고로, 최대한 많이, 31번 *shift* 합시다.

J. gahui and sousenkyo 6



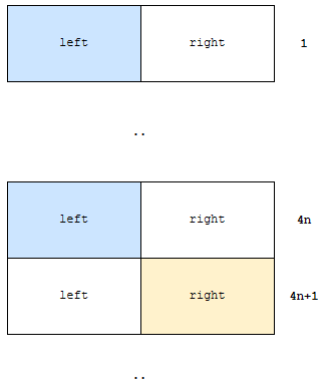
- ✓ 고로 $4 \cdot n$ 까지 *left* 부분에 5부터 $4 \cdot n'$ 을 채웁니다.
- ✓ 그 다음에는 어떻게 하면 될까요?

J. gahui and sousenkyo 6



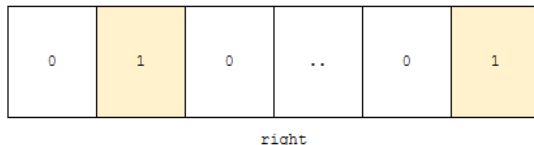
- ✓ *left* 부분은 *xor* 시키면 0으로 상쇄됩니다.
- ✓ 고로, 이 경우는 그냥 1 번째에 v 를 넣어주면 됩니다.

J. gahui and sousenkyo 6



- ✓ n 이 4로 떨어지지 않으면 어떻게 해야 할까요?
- ✓ 이 경우, 상쇄 전략을 씁니다.

J. gahui and sousenkyo 6



- ✓ x 가 위 그림과 같다고 해 봅시다.
- ✓ 그러면 어떻게 간결하게 xor 결과가 x 가 나오게 하는 두 수를 구할 수 있을까요?
- ✓ 3개의 수, 4개의 수는 어떻게 구할까요?

J. gahui and sousenkyo 6

X	1	1	1	..	1	1
Y	1	1	1	..	1	1
	right					

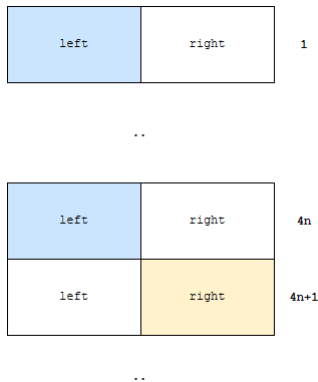
- ✓ X 와 Y 를 봅시다.
- ✓ X 에서는 군청색 부분만 보면 되고, Y 도 마찬가지입니다.
- ✓ 먼저, X 에 적절히 군청색 부분을 추가해 $mask$ 를 만들었습니다.

J. gahui and sousenkyo 6

X	1	1	1	..	1	1
Y	1	1	1	..	1	1
	right					

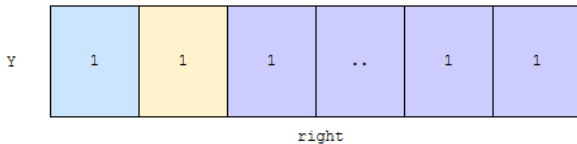
- ✓ Y 는 *mask* 입니다.
- ✓ X 의 군청색 부분과 Y 의 군청색 부분을 *xor* 하면 뭐가 나오나요?
- ✓ 당연히도 노란색 부분이 남습니다.

J. gahui and sousenkyo 6



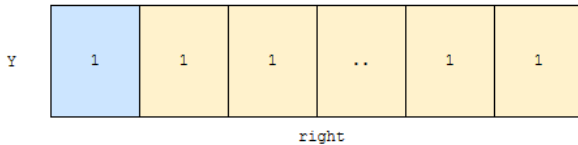
- ✓ 고로 위 그림에서 X 는 1번째 수에 더하고
- ✓ Y 는 적절히 분배해서 노란색에 넣으면 됩니다.

J. gahui and sousenkyo 6



✓ $n = 3$ 일 때 비트를 분배하는 방법 중 하나입니다.

J. gahui and sousenkyo 6



✓ $n = 2$ 일 때 비트를 분배하는 방법 중 하나입니다.

G. 가희와 집

`data_structure`, `prefix_sum`

출제진 의도 – **Challenge**

- ✓ 제출 7번, 정답 2명 (정답률 28.571%)
- ✓ 처음 푼 사람: **dablller**, 118분
- ✓ 출제자: chogahui05

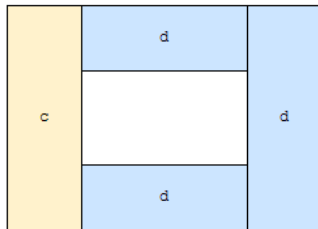
G. 가희와 집

- ✓ 요약하면, 조건에 맞는 m 자 부분합의 최소를 구하라는 문제입니다.
- ✓ 직사각형 부분합의 최소는 **금광세그**로 익숙하신 거 같습니다.
- ✓ 이 문제는 어떨까요? **금광세그** 문제일까요?

G. 가희와 집

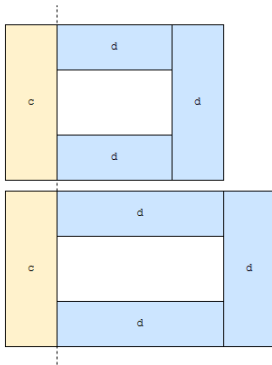
- ✓ 밑변 하나가 고정되어 있기 때문에
- ✓ 높이를 변화시키면서 무언가를 하면 된다는 것을 알 수 있어요.
- ✓ 각 높이에 대해 *sliding_window* 등을 돌리면 될 거 같습니다. 여기까지는 쉽습니다.
- ✓ 그런데 저는 $p2$ 이상으로 평가했습니다.

G. 가희와 집



- ✓ 먼저 \square 을 분해한다는 아이디어가 필요합니다.
- ✓ 왼쪽 세로와 역 Γ 자로 분해했습니다.
- ✓ 왜 이렇게 분해할까요?

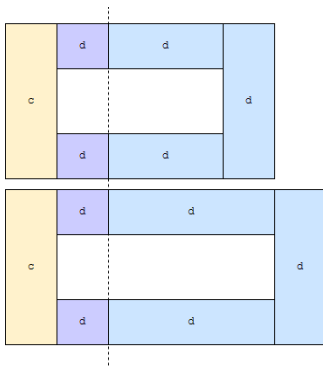
G. 가희와 집



- ✓ 세로 축이 이동하면
- ✓ 가능한 \square 자들은 하나의 세로축을 공유하기 때문입니다.
- ✓ 역 \sqsubset 자는 자료구조로 관리하면 됩니다.

가희와 함께 하는 6회 코딩테스트

G. 가희와 집



- ✓ 세로 축이 왼쪽으로 1칸 이동하면
- ✓ 자료구조에 있는 모든 정보들이 k 만큼 더해집니다.

G. 가희와 집

- ✓ 세로 축이 하나 이동하면 어떤 일이 벌어질까요?
 - 자료구조에 있는 모든 정보들이 k 만큼 더해집니다.
 - 새로운 역 \sqsubset 자 정보가 추가됩니다.
- ✓ 그런데 자료구조에 있는 모든 정보에 k 를 더하면 어떨까요?
- ✓ 21773 번에서도 비슷한 상황이 있지 않았나요?
- ✓ 대신에 역 \sqsubset 자를 넣을 때 k 를 빼면 어떨까요?

G. 가희와 집

- ✓ 저는 A 보다 10점 더 높습니다.
- ✓ 그러면 A 는 저보다 -10 점 높습니다.
- ✓ 이것은 **상대적인** 개념입니다.
- ✓ 어느 것을 기준으로 하는 것이 중요한 것입니다. 이를 이 문제에서 적용해 봅시다.

G. 가희와 집

3		
5	2	
8	5	6

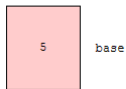
- ✓ 자료구조에 3을 넣었습니다.
- ✓ 자료구조에 있는 모든 수에 2를 더하고 3을 넣었습니다.
- ✓ 다시 자료구조에 있는 모든 수에 3을 더하고 6를 넣었습니다.

G. 가희와 집

3		
3	2-2	
3	2-2	6-2-3

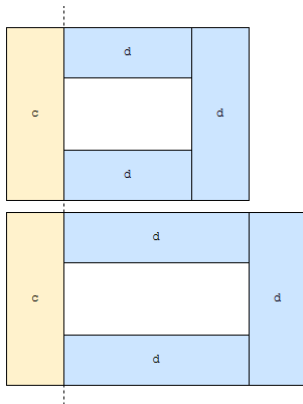
- ✓ 상대적인 관점에서 보면 이전 페이지의 상황과
- ✓ 이 페이지의 상황이 같나요? 그런 것 같습니다.

G. 가희와 집



- ✓ 그러면 원래 상황으로는 어떻게 복원하면 될까요?
- ✓ 원래 값에서 5만큼 빠진 값이 들어갔으니까 복원할 때에는 5를 더하면 됩니다.

G. 가희와 집



- ✓ 역 ㄷ자에서 최소값을 찾았습니다.
- ✓ 노란색은 값이 고정이므로 역 ㄷ의 최소값에 노란색을 더하면 됩니다.

가희와 함께 하는 6회 코딩테스트

G. 가희와 집

- ✓ 이제 *sliding_window* 를 할 때
 - 역 ㄷ자 정보를 넣습니다.
 - 자료구조에 $w_2 - w_1 + 1$ 개보다 많이 들어갔다면
 - 가장 먼저 들어간 정보를 제거하면 됩니다. 그리고 최소값 찾으면 됩니다.
 - *fifo* 특성을 가지는 *queue* 로 관리를 하면 보다 쉽습니다.

G. 가희와 집

- ✓ 이 자료구조를 *map* 등의 균형 이진트리로 관리하면 됩니다.
- ✓ 당연하게도 *key*가 중복되는 경우 처리를 잘 해주셔야 합니다.
- ✓ 시간 복잡도는 $\mathcal{O}(HW \log W)$ 입니다.

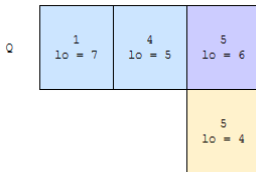
G. 가희와 집

- ✓ *map*과 *set* 등의 *tree* 기반 자료구조가 느리다면 어떤 방법이 있을까요?
- ✓ 범위 내에서 최소, 최대값만 찾는다면 *priority_queue*도 유용합니다.
 - 역 ㄷ자 정보를 넣을 때, 어느 위치가 오른쪽인지 넣고
 - 최소값을 탐색할 건데요. 너비가 w_2 보다 크다면 계속 제거하면 됩니다.
 - 그리고 최소값을 가져오면 됩니다.

G. 가희와 집

- ✓ $O(HW)$ 에 하는 방법은 없을까요?
- ✓ 놀랍게도 이 문제에서도 *monotonic_queue*를 쓸 수 있습니다. 왜?
- ✓ 일정 방향으로만 축이 이동하기 때문입니다.

G. 가희와 집



- ✓ 왼쪽으로 이동한다면 lo 는 계속 감소할 겁니다.
- ✓ 만약에 $w2$ 가 3이라고 합시다. 그러면
 - 그러면 $lo = 6$ 인 것은 3까지 영향을 미칩니다.
 - 하지만, $lo = 4$ 인 것은 1까지 더 늦게까지 영향을 미칩니다.

G. 가희와 집

Q

1 $lo = 7$	4 $lo = 5$	5 $lo = 4$
---------------	---------------	---------------

- ✓ 고로, 값이 5인 정보는 $lo = 4$ 로 대체됩니다.
- ✓ 이제 위치가 3일 때를 생각해 봅시다.

G. 가희와 집

Q

1 lo = 7	4 lo = 5	5 lo = 4
-------------	-------------	-------------

- ✓ 유효하지 않는 정보를 제거합니다.
- ✓ $7 - 3 = 4 > 3$ 이니 이것만 유효하지 않습니다.
- ✓ 따라서 최소값은 4가 됩니다.

G. 가희와 집

- ✓ *monotonic_queue* 는 이 회차에 처음 등장했지만
- ✓ 비슷한 논리를 쓰는 문제는 의외로 많이 났습니다.
- ✓ 의미 없는 정보는 버린다는 논리는 중요하니 알아두시면 좋습니다.