

서강대학교  
청정수컵 2022

Official Solutions



새내기 Round | 청정수 Round

주관



후원





# 새내기 Round

문제	의도한 난이도	출제자
<b>A</b> 두~~부 두부 두부	<b>Beginner</b>	박준서 <sup>p_jun</sup>
<b>B</b> 청정수열 (Easy)	<b>Beginner</b>	강효규 <sup>djs100201</sup>
<b>C</b> 서강의 역사를 찾아서	<b>Easy</b>	김성현 <sup>dart</sup>
<b>D</b> 두라무리 휴지	<b>Easy</b>	손기령 <sup>lickelon</sup>
<b>E</b> 배스킨라빈스~ N~ 귀엽고~ 깜찍하게~	<b>Medium</b>	박준서 <sup>p_jun</sup>
<b>F</b> 썸 팰린드롬	<b>Medium</b>	이재혁 <sup>duram21</sup>
<b>G</b> Swap the elements	<b>Hard</b>	강효규 <sup>djs100201</sup>
<b>H</b> 청정수열 (Hard)	<b>Challenging</b>	강효규 <sup>djs100201</sup>



## A. 두~~부 두부 두부

implementation, math

출제진 의도 – **Beginner**

- ✓ 제출 101번, 정답 14명 (정답률 13.861%)
- ✓ 처음 푼 사람: **이준석**, 6분
- ✓ 출제자: 박준서<sup>p\_jun</sup>



$M$  으로부터 반시계 방향으로  $K - 3$  번째에 있는 번호를 찾는 문제입니다.

반시계 방향으로 이동하는 과정에서 번호가  $[1, N]$  를 벗어나지 않게 주의해야 합니다.

- ✓ 모듈러 연산을 이용하여  $K - 3$  을 더한 후  $[1, N]$  으로 이동시켜  $O(1)$  에 해결할 수 있습니다.
- ✓ 이동할 횟수  $K$  가 크지 않기 때문에 반복문을 이용해  $O(K)$  에 해결할 수도 있습니다.
  - 반복문을 이용할 경우,  $K$  가 음수가 될 수 있음에 유의합니다.



## B. 청정수열 (Easy)

math

출제진 의도 – **Beginner**

- ✓ 제출 20번, 정답 18명 (정답률 90.000%)
- ✓ 처음 푼 사람: **채성우**, 4분
- ✓ 출제자: 강효규<sup>djs100201</sup>

## B. 청정수열 (Easy)



청정수열의 점수가 구해지는 과정을 잘 살펴봅시다.

- ✓ 1 이상  $N$  이하인 정수  $i$ 에 대해 두 개의  $i$  사이에 있는 수의 합은 최소  $2i$ 입니다.
- ✓ 따라서 모든  $i$ 에 대해 두 개의  $i$  사이의 합이 항상  $2i$ 가 되는 수열이 존재한다면, 점수가 최소인 청정수열이라고 할 수 있습니다.

## B. 청정수열 (Easy)



- ✓  $[1, 1, 2, 2, 3, 3, \dots, N, N]$  과 같은 수열을 생각해봅시다.
- ✓ 두 개의  $i$  사이에 있는 수의 합은 항상  $2i$ 입니다.
- ✓ 각  $i$ 에 대해 두 개의  $i$ 가 연속되어 있을 때, 이를 만족한다는 사실을 알 수 있습니다.

## B. 청정수열 (Easy)



- ✓ 각  $i$ 에 대해 두 개의  $i$ 가 연속되어 있는 수열의 개수는 다음과 같이 구할 수 있습니다.
  - 두 개의  $i$ 를 하나로 묶어줍니다.
  - 이제  $N$ 개의 묶음을 배치하는 서로 다른 경우의 수를 세면 되므로 정답은  $N!$ 이 됩니다.
- ✓ 시간복잡도  $O(N)$ 으로 해결할 수 있습니다.





## C. 서강의 역사를 찾아서

implementation

출제진 의도 – **Easy**

- ✓ 제출 40번, 정답 17명 (정답률 42.500%)
- ✓ 처음 푼 사람: **도정민**, 22분
- ✓ 출제자: 김성현<sup>dart</sup>



- ✓ 어떤  $i$  번째 장소가  $i \leq N, i \leq M$  라면 현재와 예전 모두 시설이 존재했던 장소입니다.
  - 현재 있는 시설보다 예전에 있던 점수가 더 높을 경우에만 예전 상태로 되돌리는 것이 이득입니다.
  - 그러한 경우가 없으면, 되돌리지 않습니다.



- ✓ 어떤  $i$  번째 장소가  $N < i \leq M$  라면 현재 시설은 없지만 예전의 시설을 불러올 수 있습니다.
  - 현재 시설이 없는 장소에 예전 시설을 불러오게 되면 예전 시설의 점수만큼 그대로 점수 합을 증가시킬 수 있습니다.
- ✓ 어떤  $i$  번째 장소가  $M < i \leq N$  라면 현재 시설이 있지만 예전에는 시설이 없었던 경우입니다.
  - 현재 있는 시설을 예전 시설이 없던 상태로 되돌리면 점수가 감소하기 때문에 되돌리지 않습니다.
- ✓ 모든 경우에 대해 점수 증가량의 최대값을  $O(\max(N, M))$  에 찾을 수 있습니다.



## D. 두라무리 휴지

string, implementation

출제진 의도 – **Easy**

- ✓ 제출 31번, 정답 6명 (정답률 19.355%)
- ✓ 처음 푼 사람: **채성우**, 10분
- ✓ 출제자: 손기령<sup>lickelon</sup>



두 단어가 주어진 조건을 만족하는지 확인하는 문제입니다.

- ✓ 두라무리 효과의 조건을 변형하면, 더 쉽게 해결할 수 있습니다.
- ✓ 첫 번째 조건은 다음과 같은 두 조건으로 나누어 생각할 수 습니다.
  - 두 단어의 자음의 개수가 서로 같다.
  - 두 단어의 모음의 개수가 서로 같다.
- ✓ 이 중 자음에 대한 조건은 세 번째 조건에 포함되는 조건입니다.



- ✓ 따라서 다음 과정을 통해 두 단어가 두라무리 효과를 일으키는지 확인할 수 있습니다.
  - 두 단어의 처음과 마지막 글자가 같은지 확인한다.
  - 두 단어에서 모음을 제거하면서 각 모음의 수가 같은지 확인한다.
  - 모음을 제거하고 남은 단어가 같은지 확인한다.
- ✓ 모든 과정이 참이라면 두 단어는 두라무리 효과를 일으키게 됩니다.
- ✓ 여기서 모음을 제거하는 과정은 자음만으로 이루어진 새로운 문자열을 생성하는 방법으로도 가능합니다.



- ✓ 또한, 첫 번째 조건을 나누지 않고 확인하는 것도 가능합니다.
- ✓ 이 경우 판별하는 과정은 다음과 같습니다.
  - 두 단어의 처음과 마지막 글자가 같은지 확인한다.
  - 두 단어의 글자들을 정렬한 결과가 같은지 확인한다.
  - 모음을 제거하고 남은 단어가 같은지 확인한다.
- ✓ 이 역시 모든 과정이 참이라면 두 단어는 두라무리 효과를 일으키게 됩니다.
- ✓ 단, 정렬을  $O(N^2)$ 에 수행할 경우 시간 초과가 발생할 수 있음에 유의해야 합니다.



## E. 배스킨라빈스 ~N~ 귀엽고 ~깜찍하게~

math, game\_theory

출제진 의도 - **Medium**

- ✓ 제출 21번, 정답 6명 (정답률 28.571%)
- ✓ 처음 푼 사람: **채성우**, 14분
- ✓ 출제자: 박준서<sup>p\_jun</sup>





$N$  과  $M$  의 범위가 크기 때문에 규칙을 찾는 접근을 해보겠습니다.  
문제를 관찰합시다.

- ✓ 자신의 턴에  $N - 1$  을 부른다면 상대는  $N$  을 불러야만 하므로 승리할 수 있습니다.
- ✓ 상대방은 최대  $M$  개의 수를 부를 수 있고, 자신은 최소 1 개 이상의 수를 불러야 합니다.
- ✓ 따라서 이전 턴에 상대가 몇 개의 수를 불렀든 내가 부른 수와 합쳐  $M + 1$  개가 되도록 하는 것이 가능합니다.



- ✓  $N - 1$ 이  $M + 1$ 로 나누어 떨어지는 경우, 후공인 상대는 내가 부른 수와 합쳐  $M + 1$ 개가 되도록 부를 수 있습니다. 즉, 상대가 반드시 승리합니다.
- ✓ 반대로 나누어 떨어지지 않는 경우, 첫 턴에  $N - 1$ 을  $M + 1$ 로 나눴을 때의 나머지만큼 수를 부릅니다. 그 후로부터  $N - 1$ 까지  $M + 1$ 개의 수를 맞춰 부를 수 있습니다. 즉, 선공이 반드시 승리합니다.
  - $M$ 이  $N$ 보다 크다면 바로  $N - 1$ 개를 불러 반드시 승리할 수 있습니다.



## F. 섬 팰린드롬

ad\_hoc

출제진 의도 – **Medium**

- ✓ 제출 23번, 정답 8명 (정답률 34.783%)
- ✓ 처음 푼 사람: **채성우**, 19분
- ✓ 출제자: 이재혁<sup>duram21</sup>

## F. 썸 팰린드롬



- ✓ 썸 팰린드롬 수를 최소 자릿수로 만들기 위해서는 가장 큰 9를 많이 사용하는 것이 이득입니다.
- ✓ 팰린드롬 수의 앞 뒤로 자릿수 9를 추가한다면, 자리수의 합은 18씩 증가합니다.
  - 따라서 최대  $\lfloor \frac{N}{18} \rfloor$  번 추가할 수 있습니다.
- ✓ 이렇게 구성한 팰린드롬 수의 자릿수는  $\lfloor \frac{N}{18} \rfloor \times 2$ 입니다.

$$\underbrace{999 \dots 999}_{\lfloor \frac{N}{18} \rfloor} \underbrace{999 \dots 999}_{\lfloor \frac{N}{18} \rfloor}$$

## F. 썸 팰린드롬



- ✓ 이제 자릿수 합에서 모자란  $N \bmod 18$  ( $N$ 을 18로 나눈 나머지)를 살펴봅시다.
- ✓  $0 < N \bmod 18 < 10$ 인 경우는 앞에서 구성한 팰린드롬의 정가운데에 배치하면 되므로 구한 자릿수에 1을 더합니다.
- ✓  $10 \leq N \bmod 18 < 18$ 인 경우는 나머지가 홀수인 경우와 짝수인 경우로 나눠 생각해봅시다.
  - 나머지가 짝수인 경우에는 나머지를 같은 두 값으로 나눠 팰린드롬 수의 정가운데 배치할 수 있습니다. 따라서 구한 자릿수에 2를 더합니다.
  - 나머지가 홀수인 경우에는 9를 팰린드롬 수 정가운데에 배치하고 나머지를 같은 두 값으로 나눠 적절하게 배치할 수 있습니다. 따라서 구한 자릿수에 3을 더합니다.
- ✓ 경우를 나눠 썸 팰린드롬 수를 구성함으로써 최소 자릿수를 구할 수 있습니다.



## G. Swap the elements

sorting

출제진 의도 – **Hard**

- ✓ 제출 25번, 정답 6명 (정답률 12.500%)
- ✓ 처음 푼 사람: **채성우**, 78분
- ✓ 출제자: 강효규<sup>djs100201</sup>



우선 새로운 수열  $B$ 를 만들 수 없는 경우를 먼저 생각해 봅시다.

- ✓ 가장 많이 등장하는 수  $x$ 의 개수가  $\frac{N}{2}$  보다 크다면  $B$ 를 만들 수 없습니다.
  - 비둘기집 원리에 의해  $A, B$  상에서 같은 위치에 있는  $x$ 가 반드시 존재하게 됩니다.
- ✓ 그렇다면 가장 많이 등장하는 수의 개수가  $\frac{N}{2}$  보다 작거나 같다면 항상  $B$ 를 만들 수 있을까요?



항상 만들 수 있습니다. 이후 풀이를 통해 구성적으로 증명해 보겠습니다.

- ✓ 수열의  $i$  번째 원소를  $(a_i, i)$  형식의 *pair* 로 나타낸 뒤  $a_i$  를 기준으로 정렬합니다.
  - 그러면 *pair* 는  $a_i$  값이 같은 원소들이 연속하여 있게 됩니다.
- ✓ 정렬된 *pair* 들을 순회하면서  $i$  번째 *pair* 와  $i + \lfloor \frac{N}{2} \rfloor$  번째 *pair* 를 바꿔주면 됩니다.
- ✓ 실제로 정답을 구성할 때는 *pair* 의  $i$  값을 이용하여 구성해주면 됩니다.





- ✓ 가장 많이 등장하는 수의 개수가  $\frac{N}{2}$  보다 작거나 같기 때문에,  $i$  번째 *pair*의 원소 값과  $i + \lfloor \frac{N}{2} \rfloor$  번째 *pair*의 원소 값은 항상 다릅니다.
- ✓ 따라서 모든 원소의 값을 다른 값으로 교체 해 줄 수 있습니다.
- ✓  $O(N^2)$  이나  $O(N \log N)$ 의 시간복잡도를 가지는 정렬을 이용해서 문제를 해결할 수 있습니다.



## H. 청정수열 (Hard)

math

출제진 의도 – **Challenging**

- ✓ 제출 4번, 정답 1명 (정답률 25.000%)
- ✓ 처음 푼 사람: **채성우**, 54분
- ✓ 출제자: 강효규<sup>djs100201</sup>



Easy 버전과 다르게 최대 점수를 구해야 합니다.

- ✓ 수열의 서로 다른 원소  $a$  와  $b$  의 위치관계에 따라 정답에 어느정도 기여하는지 살펴봅시다.
- ✓ 이때, 1 부터  $N$  까지의 정수  $i$  에 대해,  $2i^2$  는 고정적으로 더해지므로 제외하고 생각해봅시다.
- ✓  $a$  와  $b$  는 각각 2 개씩 수열에 존재하므로 6 개의 위치관계  $(a a b b)$ ,  $(a b a b)$ ,  $(a b b a)$ ,  $(b a a b)$ ,  $(b a b a)$ ,  $(b b a a)$  가 존재합니다.
- ✓ 각 위치관계에 대해 정답이 얼마나 증가하는지 살펴봅시다.

## H. 청정수열 (Hard)



- ✓  $(a a b b)$  는 0점이 추가됩니다.
- ✓  $(a b a b)$  는  $2ab$ 점이 추가됩니다.
- ✓  $(a b b a)$  는  $2ab$ 점이 추가됩니다.
- ✓  $(b a a b)$  는  $2ab$ 점이 추가됩니다.
- ✓  $(b a b a)$  는  $2ab$ 점이 추가됩니다.
- ✓  $(b b a a)$  는 0점이 추가됩니다.
- ✓  $(a a b b)$  와  $(b b a a)$  를 제외하고, 항상 동일한 점수  $2ab$ 가 추가됩니다.

## H. 청정수열 (Hard)



- ✓ 따라서  $(a a b b)$  또는  $(b b a a)$  꼴이 하나도 나오지 않는 수열이 존재한다면, 점수가 최대인 수열이 됩니다.
- ✓ 길이  $2N$ 의 청정수열을 앞에서  $N$ 개, 뒤  $N$ 개로 이뤄진 두 개의 수열로 분리해서 생각해봅시다. 두 수열 모두 길이  $N$ 인 순열 형태인 경우, 위의 꼴이 존재하지 않음을 알 수 있습니다.
- ✓ 순열이 아니라면 어떨까요?
  - 순열이 아니라면, 앞의 수열에만 나오는 수  $a$ 와, 뒤의 수열에만 나오는 수  $b$ 가 존재합니다.
  - 이는  $(a a b b)$  꼴이 되어 점수가 최대가 아닙니다.

## H. 청정수열 (Hard)



- ✓ 따라서 전체 경우의 수는  $N! \times N!$ 이 됩니다.
- ✓ 최대 점수는 위에 해당하는 아무 수열이나 만들어 구할 수 있습니다.
- ✓ 전체 시간복잡도  $O(N)$ 에 해결할 수 있습니다.
  - integer overflow에 유의하도록 합시다.



# 청정수 Round

문제	의도한 난이도	출제자
<b>A</b> 인생은 한 방	<b>Beginner</b>	박재형 <sup>pjh6792</sup>
<b>B</b> 동가수열 구하기	<b>Beginner</b>	김동건 <sup>dong_gas</sup>
<b>C</b> 카드 뽑기	<b>Easy</b>	한상필 <sup>tksv1f123</sup>
<b>D</b> INFP 두람	<b>Easy</b>	이재혁 <sup>duram21</sup>
<b>E</b> 고인물이 싫어요	<b>Medium</b>	박재형 <sup>pjh6792</sup>
<b>F</b> 1, 3, 모 나누기	<b>Medium</b>	김정모 <sup>meque98</sup>
<b>G</b> 시니컬한 개구리	<b>Hard</b>	이상원 <sup>gumgood</sup>
<b>H</b> 피엣산 청정수	<b>Challenging</b>	손기령 <sup>lickelon</sup>



## A. 인생은 한 방

string

출제진 의도 – **Beginner**

- ✓ 제출 63번, 정답 23명 (정답률 36.508%)
- ✓ 처음 푼 사람: **이도안**, 3분
- ✓ 출제자: 박재형<sup>pjh6792</sup>





- ✓ “두 문자가 사전상에서 이웃하다”는 “아스키 코드 (ASCII Code)의 차이가 1이다”라는 뜻과 같습니다.

$$\text{abs}(s[i] - s[i + 1]) = 1$$

- ✓  $k$ 가 부분 문자열의 시작 위치일 때,  $k \leq i \leq k + 3$ 에 대해 위 조건을 만족하면 “YES”입니다.
- ✓ 따라서, 1 부터  $N - 4$ 까지의 각 부분 문자열 시작 위치에 대해 5개의 문자를 확인하면 됩니다.
- ✓ 시간복잡도  $O(N)$ 에 해결할 수 있습니다.



## B. 동가수열 구하기

constructive

출제진 의도 – **Easy**

- ✓ 제출 50번, 정답 18명 (정답률 36.000%)
- ✓ 처음 푼 사람: **조성훈**, 11분
- ✓ 출제자: 김동건<sup>dong\_gas</sup>

## B. 동가수열 구하기



- ✓ 두 조건을 만족하는 수열을 구성할 수 있는지 물어보는 문제입니다.
- ✓ 이런 유형의 문제가 재미있었다면 ‘구성적’ 혹은 ‘애드 혹’ 태그가 달린 문제들을 찾아 풀어보세요 :)



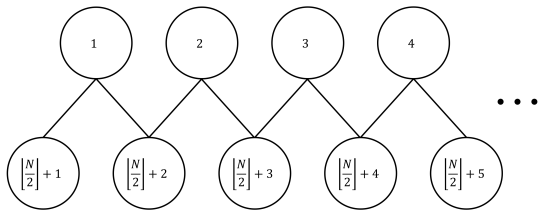
## B. 동가수열 구하기

- ✓ 두 번째 조건의  $\lfloor \frac{N}{2} \rfloor$ 에 주목해 봅시다.
- ✓ 먼저 1부터  $N$ 까지의 수를  $\lfloor \frac{N}{2} \rfloor$ 를 기준으로 두 집합으로 나누어 봅시다.
  - $1, 2, 3, \dots, \lfloor \frac{N}{2} \rfloor$
  - $\lfloor \frac{N}{2} \rfloor + 1, \lfloor \frac{N}{2} \rfloor + 2, \lfloor \frac{N}{2} \rfloor + 3, \dots, N$



## B. 동가수열 구하기

- ✓ 두 집합의 수들을 아래 그림과 같은 규칙으로 나열해 봅시다.
- ✓ 아래 그림에서 선으로 연결된 수들은 모두 차이가  $\lfloor \frac{N}{2} \rfloor$  이상입니다.
- ✓ 따라서 선으로 연결된 순서대로 수열을 구성하면 됩니다. 물론 이를 반대로 뒤집은 수열도 답이 될 수 있습니다.



## B. 동가수열 구하기



- ✓ 앞 슬라이드의 풀이대로 수열을 구성한 예시는 다음과 같습니다.
- ✓  $N$ 이 8인 경우
  - [5, 1, 6, 2, 7, 3, 8, 4]
- ✓  $N$ 이 9인 경우
  - [5, 1, 6, 2, 7, 3, 8, 4, 9]



## C. 카드 뽑기

implementation

출제진 의도 – **Easy**

- ✓ 제출 27번, 정답 7명 (정답률 25.926%)
- ✓ 처음 푼 사람: **조성훈**, 22분
- ✓ 출제자: 한상필<sup>tksvlf123</sup>



각 테스트 케이스의 카드 네 장에 대해서 아래의 규칙들 중 적어도 하나를 만족하는지 확인하면 되는 구현 문제입니다.

- ✓ 편의상 주어진 규칙을 위에서부터 순서대로 규칙 1, 2, 3이라 하겠습니다.
  - 적힌 알파벳이 같으면서 숫자가 연속되는 세 장이 존재한다.
  - 적힌 알파벳과 숫자가 모두 같은 세 장이 존재한다.
  - 두 장씩 짝지었을 때, 짝을 지은 카드끼리 적힌 숫자와 알파벳이 같다.



### C. 카드 뽑기



- ✓ 규칙 1과 2는 카드 네 장 중 세 장을 골라 확인하는 조건입니다.
  - 3중 for문을 이용하여 모든 방법으로 카드를 고를 수 있습니다.
  - 언어에 따라 next\_permutation 함수를 지원한다면 이를 이용할 수도 있습니다.
  - 혹은 세 장의 카드를 고르는 방법의 수가  $\binom{4}{3} = 4$ 로 크지 않으므로 하나씩 확인하는 방법도 있습니다.
- ✓ 규칙 3은 카드 네 장을 두 장씩 짝지어 같은지 확인하는 조건입니다.
  - 비슷한 방법으로  $\frac{\binom{4}{2}}{2!} = 3$  번이면 확인할 수 있습니다.
- ✓ 테스트 케이스 당 최대 31 번이면 확인할 수 있고,  $T$ 가 최대 100 이므로 충분히 해결할 수 있습니다.



- ✓ 정렬을 이용한다면 확인해야 하는 경우의 수를 줄이는 방법도 있습니다.
- ✓ 정렬을 이용한 풀이에서 규칙 1 확인 시 1, 2, 3 번 카드와 2, 3, 4 번 카드경우만 확인한다면 다음과 같은 경우에서 오답을 낼 수 있으니 주의해야 합니다.
  - (2m, 3m, 3m, 4m)
  - (6p, 6s, 7p, 8p)



- ✓ 다른 풀이로는 규칙을 하나라도 만족하는 모든 경우를 다 찾아놓은 뒤, 테스트 케이스마다 확인만 하는 방법이 있습니다.
- ✓ 카드를 숫자에 대응시키면 각 카드의 조합을 4차원 배열의 각 위치에 대응시킬 수 있습니다. 규칙을 하나라도 만족하면 해당 위치에 체크해두면 됩니다.
- ✓ 이때  $27^4 = 531,441$  개의 배열을 위한 공간이 필요합니다.
- ✓ 규칙을 하나라도 만족하는 경우의 수는 17,361 가지로 충분히 시간 내에 해결할 수 있습니다.



## D. INFP 두람

greedy

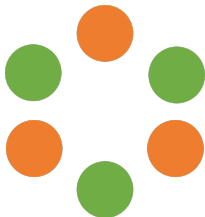
출제진 의도 – **Easy**

- ✓ 제출 46번, 정답 4명 (정답률 8.696%)
- ✓ 처음 푼 사람: **조성훈**, 75분
- ✓ 출제자: 이재혁<sup>duram21</sup>

#### D. INFP 두람



- ✓ 같은 종류의 옷을 입은 사람 사이에 다른 옷을 입은 사람이 올 수 있는 지 확인 해봅시다.
- ✓  $M$  개의 옷을 이웃하지 않게 배치하려면 옷 사이의 공간  $M$  개에 다른 종류의 옷을 배치하면 됩니다.



- ✓ 따라서 다른 종류의 옷은  $M$  개 이상이 있으면 한 종류의 옷을 이웃하지 않도록 배치할 수 있습니다.



- ✓ 그러므로 개수가 가장 많은 종류의 옷이 전체의 절반을 초과한다면 이웃하지 않도록 배치할 수 없습니다.
- ✓ 그렇지 않은 경우를 관찰 해봅시다.
- ✓ 우선 개수가 가장 많은 종류의 옷  $M$  개를 원형으로 배치합니다.
- ✓ 옷 사이의 공간을  $1, 2, 3, \dots, M$  이라고 하면 나머지 옷들은 차례대로  $1, 2, \dots, M, 1, 2, \dots$  에 넣어줍니다.
- ✓ 나머지 옷은 아래 조건을 만족하기 때문에 모든 종류의 옷을 이웃하지 않게 배치할 수 있습니다.
  - 나머지 옷이 총  $M$  개 이상이기 때문에 각 공간에는 최소 한 벌의 옷이 들어갑니다.
  - 각 종류의 옷 개수는  $M$  개 이하이기 때문에 같은 종류의 옷은 서로 다른 공간에 놓입니다.



- ✓ 하지만, 예외가 있습니다.
- ✓  $N = 1$  인 경우에 따로 처리를 해주어야 합니다.
  - $d_1 = 1$  인 경우에는 이웃하지 않도록 배치할 수 있습니다.
  - $d_1 > 1$  인 경우에는 이웃하지 않도록 배치할 수 없습니다.
- ✓ 위 예외와 integer overflow에 유의하도록 합시다.



## E. 고인물이 싫어요

graph\_traversal, disjoint\_set

출제진 의도 – **Medium**

- ✓ 제출 16번, 정답 2명 (정답률 12.500%)
- ✓ 처음 푼 사람: **조성훈**, 81분
- ✓ 출제자: 박재형<sup>pjh6792</sup>





- ✓ 특정 물탱크를 방문했을 때, 어떤 물을 얻을 수 있는지를 구해야 합니다.
- ✓ 이를 위해서는 파이프를 통해 이동 가능한 모든 물탱크를 방문해야 합니다.
- ✓ 이때 그래프 탐색을 이용하면 어떤 한 물탱크에서 얻을 수 있는 물을 시간복잡도  $O(N + M)$ 에 구할 수 있습니다.
- ✓ 이를 쿼리마다 주어지는 물탱크에 대해서 매번 구해주면 시간복잡도는  $O(Q \times (N + M))$ 으로 시간초과(TLE)를 받게 됩니다.



- ✓ 따라서, 쿼리를 받기 전에 각 물탱크마다 어떤 물을 얻을 수 있는지 미리 계산을 해두어야 합니다.
- ✓ 앞에서 그래프 탐색을 이용해 얻을 수 있는 물을 구하는 방법을 잘 생각해봅시다.
- ✓ 파이프는 양방향이기 때문에, 파이프로 연결된 물탱크들은 하나의 그룹을 이루게 됩니다.
- ✓ 그러므로, 각 그룹 내의 물탱크들에서 얻을 수 있는 물은 모두 동일합니다.



- ✓ 특정 물탱크를 시작으로 깊이 우선 탐색(DFS) 또는 너비 우선 탐색(BFS)을 수행하면서 방문한 물탱크 번호와 청정수, 고인물의 수를 각각 저장해 나갑니다.
- ✓ 더 이상 방문할 물탱크가 없는 경우, 지금까지 구한 청정수와 고인물의 수를 비교하여 방문한 물탱크들에 값을 할당해주면 됩니다.
- ✓ 방문하지 않은 남은 물탱크에 대해서 계속해서 그래프 탐색을 수행해주면  $O(N + M)$  만에 모든 물탱크에서 얻을 수 있는 물을 구하게 됩니다.
- ✓ 따라서, 시간복잡도  $O(Q + N + M)$ 에 해결할 수 있습니다.



- ✓ Union-Find를 통해서 해결할 수도 있습니다.
- ✓ 간선을 통해 서로 다른 집합(그룹)이 연결될 때, 합쳐진 집합의 root 노드에 청정수와 고인물의 수를 저장합니다.
- ✓ 그리고, 매 쿼리마다 현재 물탱크가 포함된 집합의 root 노드에 저장된 청정수의 수와 고인물의 수를 비교하여 답을 계산합니다.



## F. 1, 3, 모 나누기

dp, prefix\_sum

출제진 의도 - **Medium**

- ✓ 제출 5번, 정답 0명 (정답률 0.000%)
- ✓ 출제자: 김정모<sup>meque98</sup>



- ✓  $N$ 의 범위가 크지 않기 때문에 다양한 풀이가 존재합니다.
- ✓ 다이나믹 프로그래밍을 이용해 시간복잡도  $O(N)$  또는  $O(N^2)$ 에 해결할 수 있습니다.
- ✓ 누적합을 이용해 시간복잡도  $O(N)$  또는  $O(N^2)$ 에 해결할 수도 있습니다.



- ✓ 다이나믹 프로그래밍을 이용한 풀이를 소개해드리도록 하겠습니다.
- ✓ 다음과 같이 정의합시다.
  - $dp[i][j]$  = (배열  $A$ 의  $i$  번째 원소까지 총  $j$  개의 구간으로 나뉘었을 때, 1, 3, 5 번째 구간에 포함된 원소 합의 최대값)
- ✓  $j = 1$  인 경우,  $i$  번째 원소까지 모두 첫 번째 구간에 포함된 경우로 다음과 같이 계산할 수 있습니다.
  - $dp[i][j] = dp[i - 1][j] + A[i]$



- ✓  $j = 3, 5$ 인 경우, 두 가지 경우가 있습니다.
  - $i$ 번째 원소가 이미  $j - 1$ 번째 구간에 포함된 경우
  - $i$ 번째 원소가  $j$ 번째 구간에 포함된 경우
- ✓ 다음과 같이 계산할 수 있습니다.
  - $dp[i][j] = \max(dp[i][j - 1], dp[i - 1][j] + a[i])$





- ✓  $j = 2, 4, 6$ 인 경우, 마찬가지로 두 가지 경우가 있습니다.
  - $i$ 번째 원소가 이미  $j - 1$ 번째 구간에 포함된 경우
  - $i$ 번째 원소가  $j$ 번째 구간에 포함된 경우
- ✓ 다음과 같이 계산할 수 있습니다.
  - $dp[i][j] = \max(dp[i][j - 1], dp[i - 1][j])$



- ✓ 문제의 정답은  $n$  번째 원소까지 총 6 개의 구간으로 나뉘었을 때 최댓값인  $dp[n][6]$  가 됩니다.
- ✓  $O(N)$  개의 각 상태공간을  $O(1)$  에 계산할 수 있으므로 전체 시간복잡도는  $O(N)$  입니다.



## G. 시니컬한 개구리

bfs, graph\_traversal, graphs

출제진 의도 - **Hard**

- ✓ 제출 4번, 정답 0명 (정답률 0.000%)
- ✓ 출제자: 이상원 gumgood



개구리가 최소 한 번 이상 점프해서 집으로 가는 경로부터 경우부터 생각해봅시다.

- ✓ 개구리 밥을 무시하고 원하는 칸만큼 점프하는 경우를 **슈퍼점프**라고 하겠습니다.
- ✓ 개구리가 집으로 가는 최단 경로는 반드시 **슈퍼점프**를 포함합니다.
  - **슈퍼점프**가 필요없더라도 점프 중 하나를 **슈퍼점프**라고 간주하면 됩니다.



- ✓ 슈퍼점프로 출발하는 좌표를  $(r_1, c_1)$ , 도착하는 좌표를  $(r_2, c_2)$  라고 하면 최단 경로는 다음과 같습니다.

$$(r_f, c_f) \rightarrow \cdots \rightarrow (r_1, c_1) \rightarrow (r_2, c_2) \rightarrow \cdots \rightarrow (r_h, c_h)$$

- ✓ 상하좌우 방향으로만 점프할 수 있으므로  $r_1 = r_2$  또는  $c_1 = c_2$  입니다.
- ✓ 경로  $(r_f, c_f) \rightarrow \cdots \rightarrow (r_1, c_1)$  와 경로  $(r_2, c_2) \rightarrow \cdots \rightarrow (r_h, c_h)$  는 최단 경로입니다.



$$\underbrace{(r_f, c_f) \rightarrow \cdots \rightarrow (r_1, c_1)}_{d_1[r_1][c_1]} \rightarrow \underbrace{(r_2, c_2) \rightarrow \cdots \rightarrow (r_h, c_h)}_{d_2[r_2][c_2]}$$

- ✓ 모든  $(i, j)$ 에 대해 다음 두 값을 알고 있다고 합시다.
  - $d_1[i][j] = ((r_f, c_f)$ 에서  $(i, j)$ 로 가는 최단 경로의 길이)
  - $d_2[i][j] = ((i, j)$ 에서  $(r_h, c_h)$ 로 가는 최단 경로의 길이)
- ✓ 그러면  $(r_1, c_1)$ 에서  $(r_2, c_2)$ 로 **슈퍼점프**하면서 지나가는 최단 경로의 길이는 다음과 같습니다.

$$d_1[r_1][c_1] + \mathbf{1} + d_2[r_2][c_2]$$

- ✓ 마지막으로 모든  $(r_1, c_1), (r_2, c_2)$  쌍에 대해 위 식의 최소값을 찾으면 됩니다.



- ✓  $(r_1, c_1)$ 에서는 같은 행 또는 같은 열에 있는  $(r_2, c_2)$  로만 슈퍼점프할 수 있습니다.
- ✓ 따라서 같은 행 또는 같은 열에 있는  $d_2[r_2][c_2]$  들의 최소값을  $O(NM)$  에 전처리한다면, 어떤  $(r_1, c_1)$  에서  $O(1)$  만에 “ $(r_1, c_1)$  에서 슈퍼점프하는 최단 경로의 길이”를 알 수 있습니다.
- ✓ 전처리 후 모든  $(r_1, c_1)$  에 대한 최단 경로의 길이를  $O(NM)$  에 대해 계산하면 됩니다.



- ✓  $d_1[i][j]$  와  $d_2[i][j]$  는 BFS알고리즘으로  $O(NM)$  에 구할 수 있습니다.
  - $d_2[i][j]$  의 경우, 역간선으로 계산해야하는 것에 유의합니다.
- ✓ BFS 두 번에  $O(NM)$ , 전처리와 최단 경로 계산에  $O(NM)$  으로 전체 시간복잡도  $O(NM)$  에 해결할 수 있습니다.
- ✓ 위의 방법으로 경로를 찾을 수 없는 경우, 개구리는 집에 갈 수 없습니다.
- ✓ 개구리가 이미 집에 있는 경우, 개구리는 점프하지 않아도 됩니다.





## H. 피엣산 청정수

dp, knapsack

출제진 의도 – **Challenging**

- ✓ 제출 3번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람: **김대건**, 11분
- ✓ 출제자: 손기령<sup>lickelon</sup>



## H. 피엣산 청정수

냅색의 응용 문제입니다.

- ✓  $i$  번째 사람까지 고려했을 때 오염도가  $j$  인 경우 최대로 마실 수 있는 인원을  $dp[i][j]$  라고 합시다.
- ✓ 이 경우 dp식은 다음과 같습니다.

$$dp[i][j] = \begin{cases} dp[i-1][j] & (0 \leq j < p_i) \\ \max(dp[i-1][j], dp[i-1][j-p_i] + 1) & (p_i \leq j < p_i + c_i) \end{cases}$$

- ✓ 최종적으로 최대로 마실 수 있는 인원은 다음과 같습니다.

$$\max(dp[N][1], dp[N][2], \dots, dp[N][p_N + c_N - 1])$$

## H. 피엣산 청정수



- ✓ 단 이 문제의 경우 그대로 계산을 수행할 경우 올바른 답이 나오지 않습니다.
- ✓ 왼쪽의 예제 입력을 정렬없이 사용하여 계산을 수행한 표는 오른쪽과 같습니다.

4			$j = 0$	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
1	4	$i = 1$	0	1	0	0	0	0	0
3	4	$i = 2$	0	1	0	1	2	1	1
2	2	$i = 3$	0	1	1	2	2	0	0
5	1	$i = 4$	0	1	1	2	2	1	0

- ✓ 계산 결과 최대로 물을 마실 수 있는 인원은 2명입니다.
- ✓ 하지만 실제로는  $(2, 2) \rightarrow (1, 4) \rightarrow (3, 4)$ 의 순서대로 3명이 물을 마실 수 있습니다.



- ✓ 위의 dp식에는 사람들이 물을 일정한 순서로 마신다는 전제가 들어있습니다.
- ✓ 따라서, 입력된 순서가 최적의 순서가 아닐 경우 문제가 발생합니다.
- ✓ 위의 예제의 경우 역시,  $(1, 4) \rightarrow (3, 4) \rightarrow (2, 2)$ 의 순서로는 마실 수 없습니다.
- ✓  $(1, 4) \rightarrow (3, 4) \rightarrow \dots$  다음으로 더 마시려고 할 때, 오염도가 2 이상이기 때문입니다.
- ✓ 우선, 하나의 값을 기준으로 두 사람의 순서를 변경할 수 있는지 확인해보겠습니다.
  - $p_i$ 를 기준으로 정렬한다면, 두 원소가  $(3, 8), (5, 2)$ 와 같을 경우 문제가 발생합니다.
  - $c_i$ 를 기준으로 정렬한다면, 두 원소가  $(6, 2), (2, 3)$ 와 같을 경우 문제가 발생합니다.
- ✓ 단순히 하나의 값을 기준으로 정렬하는 것으로는 문제를 해결할 수 없음을 알 수 있습니다.

## H. 피엣산 청정수



- ✓ 이제, 데이터와 dp식의 의미에 대해 좀 더 깊게 살펴보겠습니다.
- ✓ 어떤 사람의 임계치가  $c_i$ , 오염시키는 정도가  $p_i$  라는 것은, 이 사람이 물을 마신 후에 가능한 최대 오염도가  $c_i + p_i - 1$  이라는 것을 의미합니다.
- ✓ 해당 사람이 물을 마실 수 있는 최대 오염도가  $c_i - 1$  이고, 이 상태에서 물을 마실 경우 이 오염도에  $p_i$  가 추가되기 때문입니다.
- ✓ 어떤 사람이 물을 마신 후에 가능한 최대 오염도가  $c_i + p_i - 1$  라는 사실을 이용해 봅시다.
- ✓ 반대로 생각하면, 이 사람이 물을 반드시 마시게 하려면 오염도가  $c_i + p_i - 1$  가 되기 전에는 마셔야 한다는 사실을 의미합니다.
- ✓ 이는 역 또한 성립하여, 오염도가  $c_i + p_i - 1$  가 되기 전에만 마시면 이 사람은 물을 반드시 마실 수 있습니다.



## H. 피엣산 청정수

- ✓ 최적의 경우에 물을 마실 수 있는 사람들의 집합을  $S = (s_1, s_2, \dots, s_n)$  라고 해보겠습니다.
- ✓ 집합  $S$ 의 원소를 임의로 늘어놓았을 때, 이 순서가 최적인지에 대해 생각해봅시다.
- ✓ 두 원소  $(s_{k_1}, s_{k_2})$ 를 선택했을 때  $p_{k_1} + c_{k_1} > p_{k_2} + c_{k_2}$  라면, 이는 최적의 순서가 아닙니다.
- ✓  $s_{k_1}$ 은 항상  $s_{k_2}$ 보다 나중에 물을 마실 수 있기 때문입니다.
- ✓ 어떤 두 원소  $(s_{k_1}, s_{k_2})$ 의 순서를 변경하는 것은  $s_{k_1-1}$ 까지의 결과에 영향을 주지 않습니다.
- ✓ 따라서 임의로 선택된 두 원소  $(s_{k_1}, s_{k_2})$ 가  $p_{k_1} + c_{k_1} \leq p_{k_2} + c_{k_2}$ 가 아니라면, 그리디하게 순서를 변경할 수 있습니다.
- ✓ 정리하자면,  $p_i + c_i$ 를 기준으로 정렬할 경우, 입력 데이터가 최적으로 정렬됩니다.
- ✓ 정렬이 완료되면,  $i$ 번째 사람이 물을 마셔야 하는지만 확인하는 0-1 knapsack 문제로 바뀌어 처음의 dp식을 통해 답을 구할 수 있습니다.