2024 디미고 프로그래밍 챌린지 해설

Official Solution

by

대회 운영진 일동

2024년 9월 1일

문제		의도한 난이도	출제자
A	디미고에 가고 싶어	**Easy**	ecode
B	꿈 열정 나눔	**Normal**	ian0704
C	역시 내 이세계 수열은 잘못됐다	**Normal**	ecode
D	급식 배식	**Hard**	asdfguest
E	Portal Game	**Hard**	deom
F	'한국디지털미디어고등학교'는 너무 길다	**Hard**	seungchan0325
G	Ian Sequences	**Challenging**	ian0704
H	그래프의 종착지	**Challenging**	ecode
I	어디로 갈까?	**Challenging**	asdfguest

A. 디미고에 가고 싶어!

`math` `implementation`

출제진 의도 – **Easy**

- 본 대회 – 제출 26번, 정답 13명 (정답률 50.00%)
- Open Contest – 제출 387번, 정답 200명 (정답률 53.75%)
- 처음 푼 사람(본 대회): **solomonshin** , 2분 50초
- 처음 푼 사람(Open Contest): **hanseon0303** , 1분 6초
- 출제자: ecode

A. 디미고에 가고 싶어!

- ✓ 자연수 K에 대하여 구간 $[A, B]$에 포함되는 모든 $[2K-1, 2K]$ 구간은 가로로 한 번에 촬영할 수 있습니다.
- ✓ A와 B가 주어졌을 때, 구간 $[A, B]$에 포함되는 모든 $[2K-1, 2K]$ 구간의 수와 그 외 구간의 수를 합하여 답을 구할 수 있습니다.

B. 꿈 열정 나눔

`sorting` `greedy`

출제진 의도 – **Normal**

- 본 대회 – 제출 55번, 정답 10명 (정답률 18.18%)
- Open Contest – 제출 333번, 정답 161명 (정답률 50.15%)
- 처음 푼 사람(본 대회): **solomonshin** , 11분
- 처음 푼 사람(Open Contest): **jjang36524** , 4분 19초
- 출제자: ian0704

B. 꿈 열정 나눔

- 재원이의 꿈, 열정, 나눔 스탯의 합보다 세 스탯의 합이 작은 학생들 중에 가장 스탯의 합이 큰 최대 M 명의 학생들을 선정하면 됩니다.
- 팀원의 수는 재원이를 **포함하여** M 명 이하가 되어야 함에 유의합니다.

B. 꿈 열정 나눔

- 먼저, 재원이를 포함한 $N+1$ 명의 학생들 중 재원이보다 세 스탯의 합이 작은 학생들만을 차례대로 추가하여 세 스탯의 합이 내림차순이 되도록 정렬합니다.
- 이때, 정렬은 안정 정렬 (stable sort)를 사용하거나, 스탯의 합이 같은 경우 번호가 작은 학생이 먼저 오도록 합니다.
- 이렇게 하면, 첫 M 명의 학생들의 번호를 출력하면 답이 됩니다. (재원이는 무조건 포함됩니다.)
- 이 문제에서 N 의 제한이 $1 \leq N \leq 200,000$ 이므로, $\mathcal{O}(NlogN)$ 에 작동하는 합병 정렬과 같은 정렬 알고리즘을 사용해야 합니다.



C. 역시 내 이세계 수열은 잘못됐다

`math` `greedy`

출제진 의도 – **Normal**

- 본 대회 – 제출 42번, 정답 9명 (정답률 21.43%)
- Open Contest – 제출 322번, 정답 79명 (정답률 25.16%)
- 처음 푼 사람(본 대회): **solomonshin** , 21분
- 처음 푼 사람(Open Contest): **chmpro** , 4분 26초
- 출제자: ecode

C. 역시 내 이세계 수열은 잘못됐다

- 해당 문제는 $+K$ 와 $+1$ 연산을 이용해 모든 $i\,(1 \le i \le \left\lceil \frac{N}{2} \right\rceil)$ 에 대해 $A_i = A_{N-i+1}$ 이 성립하도록 하는 문제입니다.
- $A_i = A_{N-i+1}$ 이 성립하도록 하는 것은 다시 말해 $max(A_i, A_{N-i+1}) - min(A_i, A_{N-i+1})$ 의 값을 0 으로 만드는 것과 같습니다.

C. 역시 내 이세계 수열은 잘못됐다

- ✓ 지금부터 $max(A_i, A_{N-i+1}) - min(A_i, A_{N-i+1})$ 을 d 로 정의하겠습니다.
- ✓ A_i 와 A_{N-i+1} 중에서 어디에든 $+K$ 와 $+1$ 연산을 실행할 수 있으므로 d 를 0 으로 만들기 위해 d 에 $+K, +1, -K, -1$ 의 4 가지 연산을 가할 수 있다고 할 수 있습니다.

C. 역시 내 이세계 수열은 잘못됐다

- 따라서 d가 음수가 될 때까지 $-K$ 연산을 실행하고 d가 0이 될 때까지 $+1$ 연산을 실행하는 것과 d가 음수가 되지 않도록 $-K$ 연산을 실행하고 d가 0이 될 때까지 $+1$ 연산을 실행하는 것 중에서 연산 횟수가 더 적은 것의 연산 횟수를 전체 연산 횟수에 더해주면서 해결할 수 있습니다.
- d에 대한 연산 횟수 계산을 $\mathcal{O}(1)$로 처리할 수 있고, 총 $\left\lceil \frac{N}{2} \right\rceil$ 번 d를 0으로 만드는 과정을 진행하게 됩니다.
- 즉, 시간복잡도는 $\mathcal{O}(\left\lceil \frac{N}{2} \right\rceil)$ 입니다.

D. 급식 배식

dp

출제진 의도 – **Hard**

- 본 대회 – 제출 45번, 정답 4명 (정답률 8.889%)
- Open Contest – 제출 133번, 정답 28명 (정답률 21.05%)
- 처음 푼 사람(본 대회): **j2ssicaalt** , 53분
- 처음 푼 사람(Open Contest): **jjang36524** , 12분
- 출제자: `asdfguest`

D. 급식 배식

- 이 문제는 인접한 번호의 학생이 같은 음식을 배식받을 수 없다는 제약 하에
- 모든 학생들의 행복도 합이 최대가 되도록 음식을 배식하는 문제입니다.
- 학생 기준으로 본다면, j 번 학생의 i 번 음식 배식 여부가 $j+1$ 번 학생의 음식 배식에 영향을 줍니다.
- 하지만 음식 기준으로 본다면, i 번 음식이 j 번 학생에게 배식되는지 여부가 i 가 아닌 다른 음식에는 어떠한 영향도 주지 않습니다.
- 따라서 이 문제는 각 음식에 대해서 독립적으로 해결될 수 있습니다.

D. 급식 배식

- i번 음식에 대해, 해당 음식이 배식 가능한 학생들을 $a_1, a_2, \cdots, a_{L_i}$, 각 학생들의 i번 음식을 배식 받을때의 행복도 증가량을 $v_1, v_2, \cdots, v_{L_i}$ 라고 하겠습니다. (단, $a_1 < a_2 < \cdots < a_{L_i}$)
- 그리고 다음과 같이 dp 를 정의해 봅시다.
- $dp_{k,0}$: $a_1, a_2, \cdots, a_k$ 의 학생들에 대해서, a_k **학생에게 i번 음식을 항상 배식하지 않고** 나머지 학생들에게 i번 음식을 적절히 배식할때의 행복도 합의 최댓값
- $dp_{k,1}$: $a_1, a_2, \cdots, a_k$ 의 학생들에 대해서, a_k **학생에게 i번 음식을 항상 배식하고** 나머지 학생들에게 i번 음식을 적절히 배식할때의 행복도 합의 최댓값

- 그러면 다음과 같이 점화식을 세울 수 있습니다.
- $\mathrm{dp}_{k,0} = \mathbf{max}\{\mathrm{dp}_{k-1,0}, \mathrm{dp}_{k-1,1}\}$
- $\mathrm{dp}_{k,1} = v_k + \begin{cases} \mathrm{dp}_{k-1,0} & \text{if } a_{k-1}+1 = a_k \\ \mathbf{max}\{\mathrm{dp}_{k-1,0}, \mathrm{dp}_{k-1,1}\} & \text{if } a_{k-1}+1 \neq a_k \end{cases}$
- 최종적으로 i번 음식에 대한 학생들의 최대 행복도는 $\mathbf{max}\{\mathrm{dp}_{L_i,0}, \mathrm{dp}_{L_i,1}\}$로 계산할 수 있습니다.

D. 급식 배식

- 이제 알고리즘의 시간복잡도에 대해 생각해 봅시다.
- 특정한 음식에 대해서 dp 를 계산하는데 $O(L_i)$ 가 걸립니다.
- 모든 음식을 탐색하는데 $O(N)$ 이 걸리므로 모든 음식에 대해 dp 를 $O(N + \sum_{i=1}^{N} L_i) = O(N + \sum_{i=1}^{N} l_i)$ 만에 계산할 수 있습니다.

E. Portal Game

`0-1 bfs`
출제진 의도 – **Hard**

- 본 대회 – 제출 18번, 정답 2명 (정답률 11.11%)
- Open Contest – 제출 105번, 정답 34명 (정답률 33.33%)
- 처음 푼 사람(본 대회): **pdw0412** , 60분
- 처음 푼 사람(Open Contest): **fermion5** , 13분
- 출제자: deom

E. Portal Game

- 직선 경로를 그래프로 표현해봅시다.
- 문제 상황에서 칸의 종류에 따라 행동할 수 있는 방법의 종류는 3가지 입니다.
- 포탈의 시작지점이 없는 칸에는 x 번 칸에서 $x+1$ 번 칸으로 가중치가 1인 단방향 간선이 존재한다고 볼 수 있습니다.

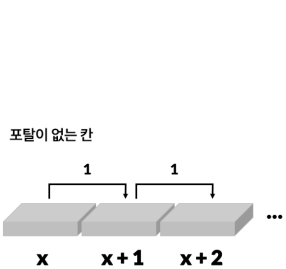
E. Portal Game

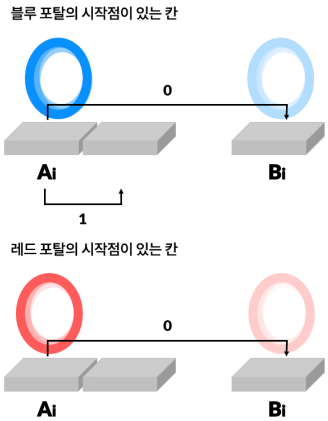
- 블루 포탈의 시작지점이 위치한 칸에서는 a_i 번 칸에서 b_i 번 칸으로 가중치가 0 인 단방향 간선이 구성될 수 있습니다.
- 블루 포탈이 위치한 칸에서는 포탈을 이용하거나 이용하지 않거나로 나뉠 수 있기 때문에 a_i 번 칸에서 $a_i + 1$ 번 칸으로 가중치가 1인 단방향 간선이 동시에 나타날 수 있습니다.

E. Portal Game

- 레드 포탈이 위치한 칸에는 반드시 레드 포탈을 이용해야 하기 때문에 a_i 번 칸에서 $a_i + 1$ 번 칸으로의 간선은 추가될 수 없고, a_i 번 칸에서 b_i 번 칸으로 가중치가 0 인 단방향 간선이 구성될 수 있습니다.
- 이렇게 직선 경로에서 가능한 이동 방법을 그래프로 변환하면 문제가 요구하는 것이 0 번 정점에서 $N - 1$ 번 정점까지의 최단거리를 구하는 것으로 바뀌게 됩니다.

E. Portal Game





E. Portal Game

- 간선이 가지는 가중치의 종류는 0 과 1 뿐이기 때문에 0-1 bfs 또는 dijkstra를 이용해서 $N-1$ 번 칸에 도착하기 위한 최소 시간을 구할 수 있습니다.
- 0-1 bfs로는 $\mathcal{O}(N)$, dijkstra로는 $\mathcal{O}(NlogN)$로 구할 수 있습니다.

F. ‘한국디지털미디어고등학교’는 너무 길다

`lcs` `ad-hoc`

출제진 의도 – **Hard**

- 본 대회 – 제출 1번, 정답 0명 (정답률 00.00%)
- Open Contest – 제출 59번, 정답 14명 (정답률 23.73%)
- 처음 푼 사람(본 대회): **N/A**
- 처음 푼 사람(Open Contest): **jjang36524** , 24분
- 출제자: seungchan0325

F. ‘한국디지털미디어고등학교’는 너무 길다

- 입력으로 주어진 문자열 S를 앞, 뒤로 연속하게 나눈 문자열을 각각 A, B라고 합시다.
- $\mathbf{min}(|A|, |B|) - \mathbf{LCS}(A, B)$를 구해야 합니다.
- 결론부터 말하면 $|A|, |B|$의 차이를 최소로 하는 것이 최적의 전략입니다.

F. ‘한국디지털미디어고등학교’는 너무 길다

- ✓ $|A| < |B|$ 라고 가정합시다. A 의 문자 하나를 B 로 옮겨 봅시다.
- ✓ A 의 문자 하나를 B 로 옮기면 A 의 길이가 1 감소하고 B 의 길이가 1 증가하기 때문에 $\mathbf{min}(|A|, |B|)$ 는 1 감소합니다.

- $\mathbf{LCS}(A, B)$는 4가지 경우로 나눌 수 있습니다.
  - 옮겨진 문자가 기존의 LCS와 새로운 LCS에 포함되지 않는다. 이 경우 $\mathbf{LCS}(A, B)$의 변화량은 0이다.
  - 옮겨진 문자가 기존의 모든 LCS에 포함되며, 새로운 LCS에는 포함되지 않는다. 이 경우 $\mathbf{LCS}(A, B)$의 변화량은 -1이다.
  - 옮겨진 문자가 기존의 LCS에 포함되지 않으며, 새로운 LCS에 포함된다. 이 경우 $\mathbf{LCS}(A, B)$의 변화량은 1이다.
  - 옮겨진 문자가 기존의 LCS와 새로운 LCS에 포함된다. 이 경우 $\mathbf{LCS}(A, B)$의 변화량은 0이다.

F. '한국디지털미디어고등학교'는 너무 길다

- ✓ 결과적으로 $\mathbf{min}(|A|, |B|) - \mathbf{LCS}(A, B))$ 의 값은 유지되거나 1 또는 2 감소합니다.
- ✓ 따라서 $|A|$ 와 $|B|$ 의 차이가 최소가 되는 것이 최적임을 알 수 있습니다.

F. '한국디지털미디어고등학교'는 너무 길다

- $|S|$가 짝수라면 S의 앞 절반을 A, 나머지를 B로 하여 계산합니다.
- $|S|$가 홀수라면 $|A|+1=|B|$인 경우와 $|A|=|B|+1$인 경우의 우열을 가릴 수 없기 때문에 두 경우 모두 계산한 뒤 더 큰 값을 택합니다.
- 일반적으로 **LCS**의 시간 복잡도는 $\mathcal{O}(|A||B|)$으로 알려져 있기 때문에 제한된 시간 안에 문제를 해결할 수 있습니다.

G. Ian Sequences

`constructive` `ad-hoc` `math`

출제진 의도 – **Challenging**

- 본 대회 – 제출 2번, 정답 1명 (정답률 100.00%)
- Open Contest – 제출 35번, 정답 25명 (정답률 71.43%)
- 처음 푼 사람(본 대회): **suhodnkla** , 124분
- 처음 푼 사람(Open Contest): **fermion5** , 23분
- 출제자: `ian0704`

G. Ian Sequences

- 이 문제는 다양한 풀이가 있지만, 대부분의 풀이가 1 부터 n 까지 모든 자연수를 두 번씩 더한 합인 $n(n+1)$ 이 $n+1$ 로 나누어 떨어진다는 점을 이용합니다.
- 이를 이용하면, 두 $n+1$ 사이에 1 부터 n 까지의 수가 두 번씩 등장하면, 두 $n+1$ 사이 수들의 합을 $n+1$ 로 나눈 나머지는 0 이 되고, 여기서 n 하나를 두 $n+1$ 사이에서 빼주면 나머지가 1 이 됩니다.

- N 이 홀수인 경우를 먼저 생각해봅시다.
- 앞에서 말한 성질을 이용하여 먼저 N 부터 1 까지의 홀수를 내림차순으로 출력하고 1 을 추가로 출력합니다. 이후 3 부터 N 까지 모든 홀수 $2i+1$ 에 대해 순서대로 $2i, 2i+1, 2i$ 를 출력해줍니다.
- 정리하자면, N 이 홀수일 때는 다음과 같은 해 구성 방식을 사용합니다.

$$N, N-2, N-4, \cdots, 3, 1, 1, 2, 3, 2, \cdots, 2i, 2i+1, 2i, \cdots, N-1, N, N-1$$

G. Ian Sequences

- N 이 짝수일 때도 비슷한 방식으로 수열을 구성할 수 있습니다.
- 먼저 N 부터 2 까지의 짝수를 내림차순으로 구성하고, 이후 2 부터 N 까지 모든 짝수 $2i$ 에 대해 순서대로 $2i-1, 2i, 2i-1$ 를 추가해줍니다.
- 정리하자면, N 이 짝수일 때는 다음과 같은 해 구성 방식을 사용합니다.

$$N, N-2, N-4, \cdots, 4, 2, 1, 2, 1, \cdots, 2i-1, 2i, 2i-1, \cdots, N-1, N, N-1$$



- 설명한 해 구성 방식은 예시일 뿐이며 더 참신하고 간단한 풀이가 존재할 수 있습니다.



H. 그래프의 종착지

`dp` `graph traversal` `dag`

출제진 의도 – **Challenging**

- 본 대회 – 제출 3번, 정답 0명 (정답률 00.00%)
- Open Contest – 제출 46번, 정답 9명 (정답률 19.57%)
- 처음 푼 사람(본 대회): **N/A**
- 처음 푼 사람(Open Contest): **jwpassion1** , 49분
- 출제자: ecode

H. 그래프의 종착지

- 정점의 개수 N 과 턴의 실행 횟수 T 가 매우 크기 때문에 시뮬레이션하면서 T 번 턴에 도달하는 정점을 찾는 시간복잡도 $\mathcal{O}(NT)$ 의 방식은 불가능합니다.

H. 그래프의 종착지



- 정점 v에 연결된 자식 노드들이 총 c개 있다고 가정하면 자식 노드의 번호를 배열 A에 오름차순으로 저장하여 $A_1, A_2, \cdots, A_c$와 같이 나타낼 수 있습니다.

H. 그래프의 종착지

- 이제 T 번의 턴을 진행하면서 정점 v 에는 최종적으로 M 번 방문한다고 생각해보겠습니다.
- 정점 v 에 방문할 때마다 정점 v 가 가리키는 자식 노드가 바뀌면서 c 번을 주기로 연결된 모든 자식 노드로 1번씩 이동하게 됩니다.
- 따라서 처음에 정점 v 가 A_u 를 가리키고 있었다면 A_u 부터 $M \bmod c$ 개의 자식 노드로 $\left\lfloor \frac{M}{c} \right\rfloor + 1$ 번 이동하게 되며, 나머지 자식 노드로는 $\left\lfloor \frac{M}{c} \right\rfloor$ 번 이동하게 됩니다.
- M 번의 방문이 끝나면 정점 v 는 $A_{((M+U-1) \bmod c)+1}$ 을 가리키게 됩니다.

H. 그래프의 종착지

- 정점 v에 최종 방문하는 횟수 M은 정점 v보다 상위 등급인 모든 노드의 최종 방문 횟수가 확정되었을 때 결정할 수 있습니다.
- 그러므로 각 노드의 등급순으로 위와 같은 과정을 거쳐 정점 v가 가리키는 자식 노드를 수정해줍니다.

H. 그래프의 종착지



- 문제에서 1등급 정점은 항상 한 개만 존재하며 최상위 노드라고 정의했으므로 1등급 정점에 $T-1$번 방문하도록 하여 위와 같은 과정은 최하위 노드까지 반복하면 $T-1$번째 턴이 끝난 후의 그래프의 모습을 결정할 수 있습니다.
- 위에서 결정된 그래프를 1등급 정점부터 따라가며 마지막으로 도착하는 노드의 번호를 출력하면 정답입니다.
- 모든 정점과 간선을 한 번씩 이용하게 되므로 시간 복잡도는 $\mathcal{O}(N+K)$입니다.

I. 어디로 갈까?

`greedy` `math`

출제진 의도 – **Challenging**

- 본 대회 – 제출 1번, 정답 0명 (정답률 0.000%)
- Open Contest – 제출 33번, 정답 3명 (정답률 9.091%)
- 처음 푼 사람(본 대회): **-** , -분
- 처음 푼 사람(Open Contest): **jjang36524** , 78분
- 출제자: `asdfguest`

- 이 문제를 풀기 위해 가장 먼저 해야하는 관찰은, 점수를 최대화 하는 두 정점을 왔다갔다 하는 꼴의 이동이 항상 존재한다는 것 입니다. (자세한 증명은 따로 첨부한 파일을 참고해 주세요.)
- 이를 이용하면 모든 간선에 대해서 모든 이동 횟수를 시뮬레이션 하는 $O(MK)$ 의 알고리즘을 짤 수 있습니다.



- 이제 두번째 관찰이 요구됩니다.
- 점수를 최대화 하는 두 정점을 왔다갔다 하는 꼴의 이동의 이동 횟수의 후보는 충분히 작게 좁혀질 수 있습니다.

- 다음과 같은 수열에 대해 생각해 봅시다.

$$a_n = \begin{cases} c_n & \text{if } 1 \le n \le T \\ a_{n-T} + c & \text{if } T < n \end{cases}$$

단 $c, c_1, c_2, \cdots, c_T, T$ 는 상수

- 만약 이 수열에서 특정한 연속한 구간에서 최솟/최대값을 구하라고 하면 주어진 구간의 앞에서부터 T 개, 뒤에서부터 T 개의 원소들만 확인하면 됩니다.

I. 어디로 갈까?

- 정점 a에서 시작해서 정점 b와 정점 a를 총 k번 왔다갔다 하는 꼴의 이동의 점수를 $f(k)$로 나타내겠습니다.
- $0 \leq k \leq K$에 대해 $f(k)$는 기본적으로 $T = 2R$인 a_n으로 나타낼 수 있습니다.
- 또한 k의 범위를 적절히 나눈다면 $f(k)$를 $T = 2$인 a_n으로 나타낼 수 있는 구간들로 분해할 수 있습니다.
- 그러면 나눠진 구간들에 대해 앞·뒤에서 2경우씩 확인하면 됩니다.

- 위의 원리에 따라 다음의 이동 횟수들을 확인하면 됩니다.

$$\{k|k = aR + b, a \in \{1, 2, l-2, l-1, l\}, b \in \{1-R, 2-R, -2, -1, 0\}\}$$

단, $l = \left\lceil \frac{K}{R} \right\rceil$

- 추가로, 예외 처리를 위해 $0, K-1, K$ 까지 고려해 주면 됩니다.
- 이제 확인해야 하는 이동 횟수의 경우가 상수 수준으로 작아졌기 때문에 $O(M)$ 으로 문제를 풀 수 있습니다.