

2022 ICPC Sinchon Winter Algorithm Camp Contest

Official Solutions

by

ICPC Sinchon

문제	의도한 난이도	출제자
EA 시간복잡도를 배운 도도	Easy	swoon
EB 상품의 주인은	Easy	songfox00
EC queuestack	Medium	swoon
ED btsp-small	Medium	lky7674
EE 좀비 바이러스	Hard	songfox00
EF n번째 숫자 찾기	Hard	wnstjd13245
EG 히히 못가	Challenging	gumgood

EA. 시간복잡도를 배운 도도

implementation

출제진 의도 - **Easy**

- ✓ 제출 66번, 정답 38명 (정답률 57.57%)
- ✓ first solve: **wltnjeon0119**, 4분
- ✓ 출제자: swoon

EA. 시간복잡도를 배운 도도

- ✓ 한 줄에 등장하는 for과 while의 개수를 세는 문제입니다.
- ✓ 반복문을 이용하여 구하면 됩니다.
- ✓ for과 while을 독립적으로 처리해주지 않으면 foile 같은 경우에서 문제가 생길 수 있습니다.
- ✓ 한 줄에 등장하는 문자의 개수가 N 개라 할 때, 시간복잡도는 $\mathcal{O}(CN)$ 입니다.

EB. 상품의 주인은?

sorting

출제진 의도 – **Easy**

- ✓ 제출 117번, 정답 29명 (정답률 24.78%)
- ✓ first solve: **kiles1201**, 13분
- ✓ 출제자: songfox00

EB. 상품의 주인은?

- ✓ 간단한 정렬 문제입니다.
- ✓ 과목별 배열을 만들어 점수가 높은 순으로 정렬합니다.
- ✓ 국어 1등인 학생의 번호를 먼저 저장한 후 영어, 수학, 과학 순서대로 1등을 구합니다.
 - 상품을 받지 않은 학생 중에서 1등을 구해야합니다.

EC. queuestack

ad-hoc, queue

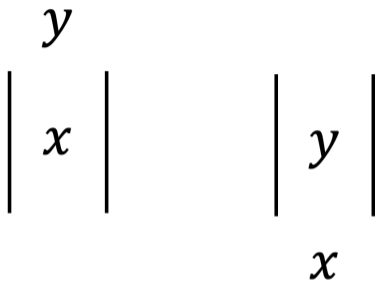
출제진 의도 – **Medium**

- ✓ 제출 104번, 정답 18명 (정답률 17.30%)
- ✓ first solve: **tonynamy**, 27분
- ✓ 출제자: swoon

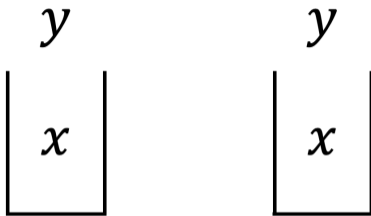
- ✓ 먼저 naive한 풀이를 떠올려봅시다.
- ✓ N 개의 자료구조를 만든 뒤, M 개의 원소를 넣었다 뺐다면 시간복잡도는 $\mathcal{O}(NM)$ 이 됩니다.
- ✓ 다른 관찰을 해 봅시다.



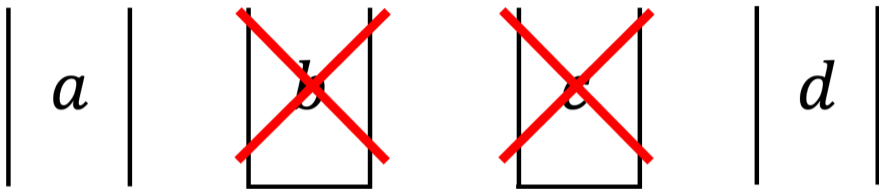
- ✓ 아래가 뚫린 자료구조는 queue, 막힌 자료구조는 stack입니다.
- ✓ 위와 같은 queuestack이 존재한다고 가정합니다.
- ✓ queue와 stack의 특성을 확인해봅시다.



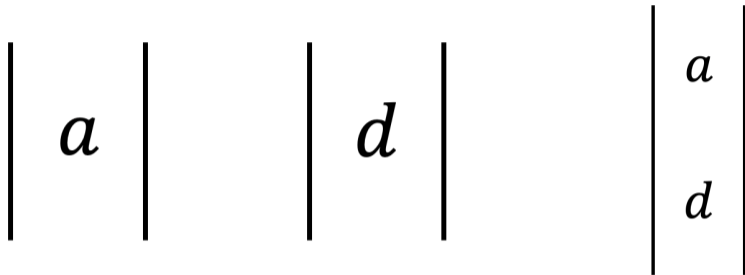
- ✓ queue에 원소 y 를 삽입한 후 pop을 해보겠습니다.
- ✓ y 가 queue 내부에 들어가게 되고 원래 있던 x 는 튀어나왔습니다.



- ✓ stack에 원소 y 를 삽입한 후 pop을 해보겠습니다.
- ✓ x 는 그대로 stack에 남아있게 되고 y 는 튀어나왔습니다.
- ✓ 이는 stack은 아무 영향도 주지 못한다는 것을 의미합니다.



- ✓ 따라서 우리는 아까 queuestack을 모든 stack이 없는 것으로 생각하고 풀 수 있습니다.
- ✓ 하지만 아직도 queue가 N 개라면 시간복잡도는 $\mathcal{O}(NM)$ 입니다.



- ✓ queue의 집합을 다 이어주게 된다면 하나의 queue로 볼 수 있습니다.
- ✓ 따라서 queue 하나만 사용하면 되므로, 시간복잡도는 $\mathcal{O}(N + M)$ 이 됩니다.

ED. Bottleneck Travelling Salesman Problem (Small)

brute force, backtracking

출제진 의도 – **Medium**

- ✓ 제출 9번, 정답 2명 (정답률 22.22%)
- ✓ first solve: **leeju1013**, 123분
- ✓ 출제자: 1ky7674

- ✓ 모든 정점을 방문하는 순회는 길이가 N 인 permutation으로 표현할 수 있습니다.
- ✓ 제한이 매우 작기 때문에 모든 길이가 N 인 permutation을 탐색하는 방법으로도 시간 내에 해결할 수 있습니다.
- ✓ 재귀 함수를 이용한 backtracking, 또는 next_permutation을 통해서 모든 길이가 N 인 permutation을 탐색할 수 있습니다.
 - java에는 next_permutation 함수가 없습니다. 주의 부탁드립니다.
- ✓ 정해의 시간복잡도는 $\mathcal{O}(N \cdot N!)$ 입니다.

EE. 좀비 바이러스

graph, graph traversal, bfs

출제진 의도 – **Hard**

- ✓ 제출 35번, 정답 4명 (정답률 11.42%)
- ✓ first solve: **tonynamy**, 52분
- ✓ 출제자: **songfox00**

- ✓ BFS를 이용하여 풀 수 있는 문제입니다.
- ✓ 입력으로 주어진 1번과 2번 바이러스의 위치 모두 queue에 넣고 시작합니다.
- ✓ 3번 바이러스가 만들어지는지 판단하기 위해서는 방문 시간을 저장해야 합니다.
- ✓ 3번 바이러스에 감염되었거나 치료제가 있는 마을은 건너뛰어야 합니다.

EF. n번째 숫자 찾기

math, binary search

출제진 의도 – **Hard**

- ✓ 제출 2번, 정답 0명 (정답률 0%)
- ✓ first solve: -, -분
- ✓ 출제자: wnstjd13245

- ✓ 이분탐색을 의도한 문제입니다.
- ✓ n 의 범위가 크지 않아 이분탐색 없이도 충분히 해결할 수 있습니다.

- ✓ $X_K(M + 1)$ 은 $M + 1$ 을 K 진수로 변환하여 $X_K(M)$ 의 뒤에 붙여주면 얻을 수 있습니다.
- ✓ YJ_K 의 N 번째 숫자를 얻기 위해서는 $X_K(M)$ 을 $M = 1$ 부터 $M = \sqrt{N}$ 까지 붙여주면 충분합니다.

- ✓ 먼저 $X_K(1)$ 부터 $X_K(M)$ 까지의 숫자의 개수 합을 계산해 저장해둘 배열을 만듭니다.
- ✓ $M = \sqrt{N}$ 까지 부분합 배열을 만들어줍니다.
- ✓ 이후 부분합 배열을 이분탐색하여 N 이 어떤 $X_K(L)$ 에 존재하는지 찾고 ($1 \leq L \leq \sqrt{N}$)
- ✓ 찾은 $X_K(L)$ 에서 $N - (X_K(1)$ 부터 $X_K(L - 1)$ 까지 숫자의 개수의 합) 번째 숫자를 찾아주면 됩니다.

EG. 히히못가

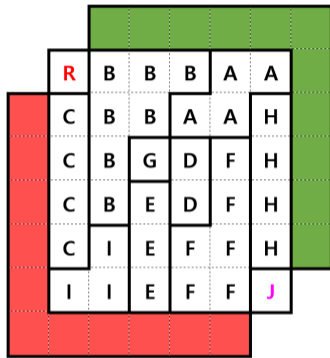
graph, dijkstra

출제진 의도 – **Challenging**

- ✓ 제출 0번, 정답 0명 (정답률 0%)
- ✓ first solve: -, -분
- ✓ 출제자: gumgood

R	B	B	B	A	A
C	B	B	A	A	H
C	B	G	D	F	H
C	B	E	D	F	H
C	I	E	F	F	H
I	I	E	F	F	J

- ✓ (1, 1)에서 (N, N)까지 상하좌우로 인접한 격자를 통해 이동하는 경로를 모두 막아야 합니다.
- ✓ 같은 영역에 속한 격자는 함께 사야 할 때, 최소 몇 개의 격자를 사야 하는지 구하는 문제입니다.



- ✓ 위와 같은 초록색 영역과 빨간색 영역을 추가합니다. 두 영역을 잇는 가장 짧은 경로에 해당하는 격자를 사면 $(1, 1)$ 과 (N, N) 사이의 경로를 모두 막을 수 있습니다.

EG. 히히 못가

- ✓ 각 영역을 노드로 두고 영역의 크기를 노드의 가중치로 설정합니다.
 - 단, 새로 추가한 초록색 영역과 빨간색 영역의 가중치는 0입니다.
- ✓ 이제 노드가 가중치를 가지는 그래프에서 최단 경로를 구하는 문제가 되었습니다.
 - 노드를 분할하거나 방문할 때마다 계산하는 방법으로 구현할 수 있습니다.
- ✓ 주의할 점은 상하좌우뿐만 아니라 네 대각선을 포함한 8방향으로 노드를 연결해야 합니다.
- ✓ Dijkstra 알고리즘으로 해결할 경우 시간복잡도는 $O(N^2 \log N)$ 입니다.

문제	의도한 난이도	출제자
HA 잘 알려진 수열 구하기	Easy	dj s100201
HB 잘 알려진 합 구하기	Medium	dj s100201
HC 카드 게임과 쿼리	Medium	swoon
HD btsp-large	Medium	lky7674
HE Meet In The Middle	Hard	gumgood
HF 스네이크 게임	Hard	gumgood
HG 토지 구입	Challenging	lky7674

HA. 잘 알려진 수열 구하기

constructive, ad-hoc

출제진 의도 – **Easy**

- ✓ 제출 39번, 정답 28명 (정답률 71.79%)
- ✓ first solve: **developerhan**, 1분
- ✓ 출제자: **djs100201**

HA. 잘 알려진 수열 구하기

- ✓ $2, 4, 6, 8 \cdots 2n - 4, 2n - 2, 2n$
- ✓ $1, 3, 5, 7 \cdots 2n - 5, 2n - 3, 2n - 1$
- ✓ 위의 수열중 하나를 출력하면 됩니다. 이외에도 여러가지 수열이 있습니다.
- ✓ 구간을 prefix sum 으로 나눠서 보면 조건을 만족함을 쉽게 보일 수 있습니다.

HA. 잘 알려진 수열 구하기

- ✓ 누적합을 저장하는 pre라는 배열을 만들어봅시다.
- ✓ 수열의 첫번째 원소 부터, i 번째 원소까지의 합을 $pre[i]$ 에 저장해봅시다.
- ✓ 수열의 i 번째 원소부터 j 번째 원소까지의 합은 $pre[j]-pre[i - 1]$ 입니다.

HA. 잘 알려진 수열 구하기

- ✓ $2, 4, 6, 8 \cdots 2n - 4, 2n - 2, 2n$ 임이 조건을 만족함을 보여봅시다.
- ✓ 그런데 위 수열의 특성상, $\text{pre}[x]=x(x+1)$ 입니다.
- ✓ $\text{pre}[j]-\text{pre}[i-1] = j^2 + j - ((i-1)^2 + i - 1) = (j+i)(j-i+1)$ 입니다.
- ✓ 수열의 길이인 $(j-i+1)$ 로 나누어 떨어지는 것을 알 수 있습니다.

HB. 잘 알려진 합 구하기

number theory, math

출제진 의도 – **Medium**

- ✓ 제출 116번, 정답 13명 (정답률 11.21%)
- ✓ first solve: **luciaholic**, 26분
- ✓ 출제자: **djs100201**

B. 잘 알려진 합 구하기

- ✓ 구간이 연속되어 있기 때문에 i 를 m 으로 나눈 나머지는 $0, 1, 2 \cdots m - 2, m - 1$ 이 반복됩니다.
- ✓ 서로 다른 몫의 값은 $\mathcal{O}(\sqrt{N})$ 개 이하이고, 효율적으로 iterate 하는 방법은 캠프 수업에서 배웠습니다.
- ✓ 같은 몫에 대한 계산을 $\mathcal{O}(1)$ 해주면 시간복잡도 $\mathcal{O}(\sqrt{N})$ 에 해결할 수 있습니다.

HC. 카드 게임과 쿼리

game, dp_bitfield

출제진 의도 – **Medium**

- ✓ 제출 33번, 정답 7명 (정답률 26.92%)
- ✓ first solve: **luciaholic**, 62분
- ✓ 출제자: swoon

- ✓ 먼저, 당연하게도 현재 남은 거리와 남은 숫자 카드를 알면 승자를 알 수 있습니다.
- ✓ 남은 거리를 i , 남은 숫자 카드를 집합 S , S 를 bitset으로 표현한 것을 j 라 두겠습니다.
- ✓ $f(i, j) =$ 남은 거리가 i 고, 숫자 카드의 bitset이 j 일 때 승리 = 1, 패배 = 0

$$f(i, j) = \begin{cases} 0, & \text{if } i \leq 0 \\ 1 \oplus \bigwedge_{t \in S} f(i - t, j - 2^t), & \text{otherwise} \end{cases}$$

- ✓ i, j 의 쌍이 최대 몇 개 나올 수 있는지 계산해보겠습니다.

- ✓ 일단 $B - A$ 즉, 거리를 N 으로 두겠습니다. 그리고 1 부터 K 까지의 합을 C 로 두겠습니다.
- ✓ 만약 N 이 C 이상이라면, $N \% C$ 까지는 게임이 강제로 흘러갑니다. 왜냐하면 모든 카드를 소진해야하기 때문입니다.
- ✓ 그러므로 우리가 신경쓸 거리의 범위는 C 미만입니다.
- ✓ 숫자카드는 K 가지이므로 총 2^K 가지의 카드 집합을 가질 수 있습니다.
- ✓ (가능한 거리의 범위) * (가능한 카드 집합의 개수) = $\mathcal{O}(C2^K)$, C 는 $K(K + 1)/2$
- ✓ 따라서 $f(i, j)$ 의 가짓수는 $\mathcal{O}(K^2 2^K)$ 입니다.

HC. 카드 게임과 쿼리

- ✓ 여기서 $f(N\%C, 2^K - 1)$ 이 1이면 swoon을 출력하고, 0일때 raararaara를 출력하게 된다면 한 가지 반례에 막히게 됩니다.
- ✓ $f(N\%C, 2^K - 1)$ 가 1이라면 저 상황에서 swoon이 승리하는 것은 맞지만, $N\%C$ 까지 어떻게 가게 됐는지도 생각해봐야 합니다.
- ✓ K 개의 카드를 모두 더한 값이 $C, N/C$ 를 R 로 두겠습니다.
- ✓ N 에서 $N\%C$ 까지 오기 위해서 플레이어는 모두 RC 번의 카드를 냅습니다.
- ✓ 여기서 R 과 C 가 만약 모두 홀수라면 원래의 시작과 선후공이 뒤바뀌게 됩니다.
- ✓ 따라서 R 과 C 가 모두 홀수인 경우 승패가 뒤바뀐다는 것을 처리해줘야 합니다.

- ✓ 각 상태를 dp table에 저장해둘 수 있고, 모든 상태는 $\mathcal{O}(K^2 2^K)$ 개 이므로 table을 채우는 공간복잡도와 시간복잡도 모두 $\mathcal{O}(K^2 2^K)$ 가 됩니다.
- ✓ 전처리를 해주면 매 쿼리당 $\mathcal{O}(1)$ 로 처리할 수 있으므로 총 시간복잡도는 $\mathcal{O}(K^2 2^K)$ 이 됩니다.

HD. Bottleneck Travelling Salesman Problem (Large)

dp_bitfield

출제진 의도 – **Medium**

- ✓ 제출 25번, 정답 6명 (정답률 40.00%)
- ✓ first solve: **meque98**, 120분
- ✓ 출제자: 1ky7674

- ✓ 제한이 small에 비해서 커졌기에 모든 길이가 N 인 permutation을 탐색하는 방법으로 시간 내에 해결할 수 없습니다.
- ✓ N 이 최대 18이기에 중급 강의에서 다뤘던 "2098 외판원 순회" 문제처럼 bitwise DP로 풀 수 있습니다.
- ✓ 시작 정점을 1로 고정하고 나면 다음과 같이 DP식을 세울 수 있습니다.
 - $dp(b, v)$: 이전에 방문한 정점의 집합을 N bits 정수로 표현한 것을 b , 현재 정점의 위치를 v 라고 했을때, 남은 정점들을 방문했을 때의 bottleneck의 최솟값.
 - $b \neq 2^N - 1$ 일 때, $dp(b, v) = \min_{2^w \wedge b=0} \max\{dp(b \vee 2^w, w), d(v, w)\}$
 - $b = 2^N - 1$ 일 때, $dp(b, v) = d(v, 1)$

- ✓ 만들어진 DP 테이블을 바탕으로 bottleneck을 최소화하도록 순회를 직접 만들어야 합니다.
- ✓ $b = 1, v = 1$ 에서 시작해서 순회를 만들어봅시다.
- ✓ $dp(b, v) = \max\{dp(b \vee 2^w, w), d(v, w)\}$, $2^w \wedge b = 0$ 를 만족하는 w 가 v 의 다음 정점이 될 수 있다. 이 과정을 $b = 2^N - 1$ 이 될 때까지 반복한다.
- ✓ 시간 복잡도 : $O(N \cdot 2^N)$

HE. Meet In The Middle

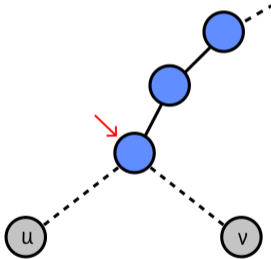
graph, lca

출제진 의도 - **Hard**

- ✓ 제출 28번, 정답 5명 (정답률 17.24%)
- ✓ first solve: **neogate**, 78분
- ✓ 출제자: gumgood

- ✓ 간선이 가중치를 가지는 트리가 주어집니다.
- ✓ 어떤 두 노드 u, v 로부터 같은 거리에 있는 노드를 찾는 문제입니다.
 - 그러한 노드가 여러 개라면, 각 노드까지의 거리 합이 가장 짧은 노드를 구해야 합니다.

- ✓ 정답에 해당하는 노드는 항상 u 와 v 를 잇는 경로 위에 있습니다.
- ✓ u, v 와 같은 거리에 있는 노드가 여러 개 있더라도, 경로 위에 있는 것이 각 노드까지의 거리 합이 가장 짧습니다.



이제 u 와 v 를 잇는 경로 상에서 두 노드까지의 거리가 같은 노드를 찾아봅시다.

- ✓ 먼저 u 에서 v 까지의 거리 l 을 구해야 합니다.
- ✓ 어떤 임의의 노드를 루트로 정한 뒤, 모든 노드 i 에서 루트까지 거리 $dep(i)$ 를 DFS 또는 BFS로 구합니다.
- ✓ 그러면 l 은 $dep(u) + dep(v) - 2 \times dep(\text{LCA}(u, v))$ 로 계산할 수 있습니다.
- ✓ 일반성을 잃지 않고, u 가 v 보다 루트로부터 멀리 있다고 하겠습니다.
 - l 이 홀수라면 답이 존재하지 않습니다.
 - l 이 짝수라면 u 에서 루트까지 $l/2$ 만큼 binary lifting을 통해 $O(\log n)$ 에 올라가서 확인해 보면 됩니다.

- ✓ LCA를 sparse table을 이용해 구현하면 binary lifting도 어렵지 않게 구현할 수 있습니다.
- ✓ 각 쿼리당 $O(\log N)$ 에 처리할 수 있으므로 전체 시간복잡도 $O((N + K) \log N)$ 에 해결할 수 있습니다.

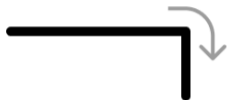
HF. 스네이크 게임

string, kmp, geometry

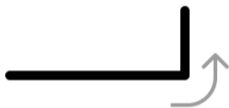
출제진 의도 – **Hard**

- ✓ 제출 4번, 정답 1명 (정답률 25%)
- ✓ first solve: **luciaholic**, 215분
- ✓ 출제자: gumgood

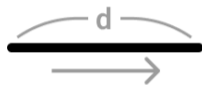
- ✓ 2차원 평면상에서 선분들의 집합을 패턴 매칭하는 문제입니다.
- ✓ 이를 문자열 패턴 매칭 문제로 변환하는 게 문제의 핵심이 되겠습니다.



-1

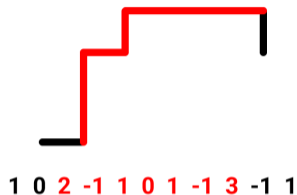
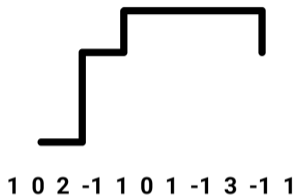


0

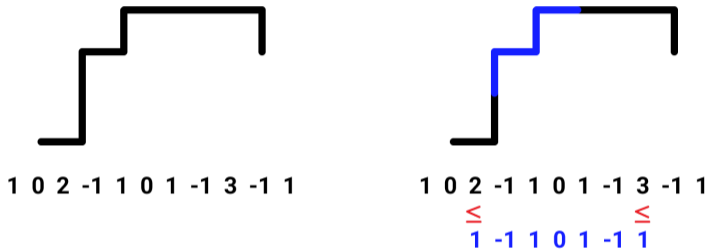


d

- ✓ 뱀을 한쪽 끝부터 시작해서 각 점에서 우회전하면 -1 , 좌회전하면 0 으로 나타내고 거리 d 만큼 직진하면 정수 d 로 나타냅니다.



- ✓ 이를 통해 뱀모양을 $[-1, 2e9]$ 범위 내 정수를 원소로 가지는 정수 배열로 나타낼 수 있습니다.
 - 트리거 패턴도 동일한 방법으로 나타낼 수 있습니다.
- ✓ 이제 패턴 매칭 알고리즘을 적용해 봅시다.



- ✓ 먼저, 양 끝 직진 구간을 제외한 뒤 패턴 매칭 알고리즘으로 일치하는 위치를 모두 구합니다.
- ✓ 각 위치들에서 제외한 직진 구간도 포함되는지 확인하여 트리거 패턴을 모두 찾을 수 있습니다.

- ✓ 마지막으로 트리거 패턴의 반대쪽 끝부터 시작해서 위의 과정을 동일하게 반복하면 됩니다.
 - 만약 반대쪽 끝부터 시작한 수열이 기존 수열과 같다면 반복할 필요가 없습니다.
- ✓ KMP 알고리즘으로 패턴 매칭을 할 경우, 시간복잡도 $O(N + M)$ 에 해결할 수 있습니다.

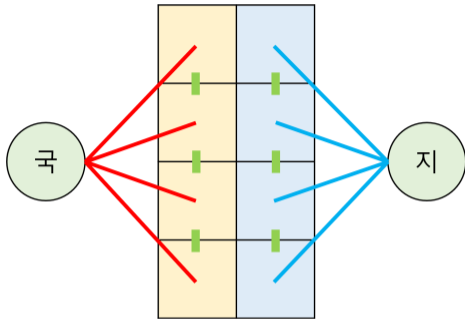
- ✓ 이 풀이를 응용하면, 뱀이 수직뿐만 아니라 다양한 각도로 회전해 있더라도 패턴 매칭을 할 수 있습니다.
- ✓ 회전을 나타내는 $-1, 0$ 대신 어떤 값을 넣으면 좋을지 생각해 봅시다.

HG. 토지 구입

flow, min cut

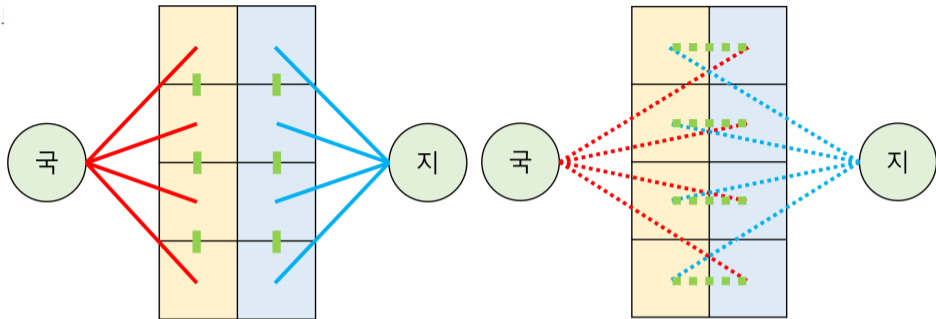
출제진 의도 – **Challenging**

- ✓ 제출 1번, 정답 1명 (정답률 100.00%)
- ✓ first solve: **ystaeyoon113**, 229분
- ✓ 출제자: lky7674



✓ 다음 이익의 합을 최대화 하는 문제입니다.

- 국령이가 땅 i 를 구입했을 때의 이익 a_i .
- 지환이가 땅 i 를 구입했을 때의 이익 b_i .
- 이웃한 땅 i, j 의 주인이 동일할 때 얻는 이익 $s_{i,j}$.



- ✓ 이익의 합을 최대화 하는 문제는 언지 못한 이익의 합을 최소화 하는 문제로 바꿀 수 있습니다.
 - 국령이가 땅 i 를 구입했을 때의 이익 a_i .
 - 지환이가 땅 i 를 구입했을 때의 이익 b_i .
 - 이웃한 땅 i, j 의 주인이 동일할 때 얻는 이익 $s_{i,j}$.
- ✓ 언지 못한 이익의 합의 최소화 → min-cut 문제로 바꿀 수 있습니다.

- ✓ 수식을 통해서 더 자세히 설명하겠습니다.
- ✓ 국령이가 산 땅의 집합: A , 지환이가 산 땅의 집합: B , 전체 땅의 집합: $X = A \cup B$.
- ✓ 얻는 이익 $\sum_{i \in A} a_i + \sum_{i \in B} b_i + \sum_{i \text{ neighbor } j, \{i,j\} \cap A \neq 1} s_{i,j}$ 을 최대화가 목적.
- ✓ 모든 이익의 합 $\sum_{i \in X} a_i + \sum_{i \in X} b_i + \sum_{i \text{ neighbor } j} s_{i,j}$ 은 A, B 상관없이 동일한 값.
- ✓ 따라서 (모든 이익의 합) - (얻는 이익) = $\sum_{i \in B} a_i + \sum_{i \in A} b_i + \sum_{i \text{ neighbor } j, \{i,j\} \cap A = 1} s_{i,j}$ 을 최소화 하는 문제로 바꿀 수 있습니다.

✓ 다음과 같은 flow network를 정의하도록 하겠습니다.

– 정점 : $\{s, t\} \cup X$

– 간선

▶ $c(s, i) = a_i$ for $i \in X$.

▶ $c(i, t) = b_i$ for $i \in X$.

▶ $c(i, j) = s_{i,j}$ for $i, j \in X$ and i neighbors j .

✓ (s-t cut $(\{s\} \cup A, \{t\} \cup B)$ 의 capacity) = $\sum_{i \in B} a_i + \sum_{i \in A} b_i + \sum_{i \text{ neighbor } j, \{i,j\} \cap A=1} s_{i,j}$

✓ 따라서 해당 flow network의 min-cut을 구하면 됩니다.