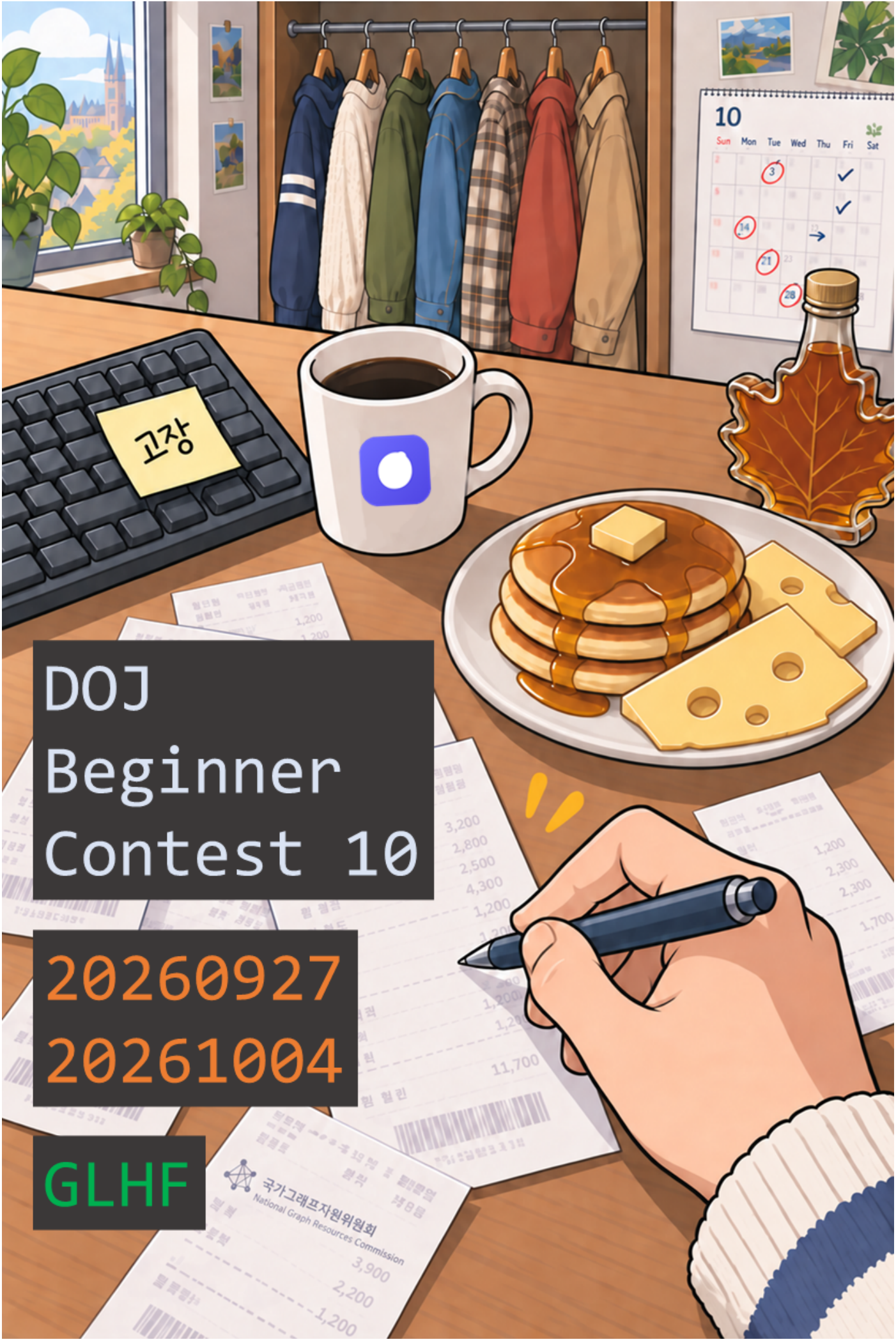


Written with GPT-6.1 Sol

문제 목차

DOJ Beginner Contest 10

문제	목표 난이도	사용 알고리즘
A. Jones rule of 86	B4	사칙연산
B. 소비세 미포함	S3	수학
C. 접촉 불량	S1 – G3	그리디 슬라이딩 윈도우 문자열 구현
D. 스위스 치즈 모델 (Easy)	G4 – G3	기하 그래프 탐색 BFS 분리집합
E. 그래프 눌러담기	G1 – P5	함수형 그래프 위상 정렬 조합론
F. 스위스 치즈 모델 (Hard)	P2?	기하
G. 패셔니스타	P2 – P1	DP

난이도는 solved.ac 티어에 맞춘 출제 의도 난이도입니다.

A. Jones rule of 86

핵심: 문제에 주어진 식에 입력값을 그대로 대입한다.

수액의 당도를 X라 하면 필요한 수액의 양은 $\frac{86}{X}$갤런이다. 실수로 입력받아 나눗셈을 한 뒤 소수점 아래 한 자리까지 반올림하여 출력하면 된다.

예를 들어 당도가 2.3이면 필요한 양은 약 37.3913이므로 37.4를 출력한다. 농도나 설탕의 양을 따로 계산할 필요는 없다.

정수 나눗셈을 하면 소수 부분이 사라지므로 실수 나눗셈을 사용한다. 또한 필요한 양을 버림하는 것이 아니라 반올림해야 하며 정수로 떨어져도 소수점 아래 한 자리를 출력한다. 예를 들어 당도가 2.0이면 출력은 43.0이다.

복잡도: 시간과 공간 모두 $O(1)$이다.

B. 소비세 미포함

핵심: 가능한 세전 가격을 전부 찾는 대신 올바른 가격이 존재하는 정수 구간을 구한다.

$a = 100 + K$라 하자. 영수증의 금액 M이 가능하려면 어떤 양의 정수 N에 대해 다음이 성립해야 한다.

$$M \le \frac{aN}{100} < M + 1.$$

양변에 100을 곱하고 aN이 정수라는 사실을 사용하면 다음과 같다.

$$100M \le aN \le 100M + 99.$$

즉 구간 안에 a의 배수가 하나라도 있으면 된다. $aN \ge 100M$을 만족하는 가장 작은 N을 구한 뒤 오른쪽 끝도 넘지 않는지 확인한다.

$$a = 100 + K, \qquad N_0 = \left\lceil \frac{100M}{a} \right\rceil = \left\lfloor \frac{100M + a - 1}{a} \right\rfloor,$$

$$\text{possible} \iff aN_0 \le 100M + 99.$$

올림은 정수 나눗셈으로 계산할 수 있다. N_0보다 작은 값은 왼쪽 조건을 만족하지 못하고 큰 값은 aN을 더 키우므로 이 후보 하나만 확인하면 충분하다. $M \ge 1$이므로 N_0도 양수이다.

이 과정에는 실수 계산이 필요 없다. `possible`이 참이면 `YES`, 거짓이면 `NO`를 출력한다.

복잡도: 영수증당 $O(1)$, 전체 시간 $O(T)$, 추가 공간 $O(1)$이다.

별해: 나눗셈 후 가까운 후보 검산. $q = \lfloor \frac{100M}{a} \rfloor$를 구한 뒤 q와 $q+1$을 원래 식에 대입해도 된다. 양의 후보 N 중 $\lfloor \frac{aN}{100} \rfloor = M$인 값이 있으면 `YES`이다. 가능한 세전 가격의 범위는 $\frac{100M}{a} \le N < \frac{100(M+1)}{a}$이고 그 길이가 $\frac{100}{a} \le 1$이므로 두 후보만 확인하면 충분하다. 이 방법도 영수증당 $O(1)$이다.

C. 접촉 불량

핵심: 같은 괄호가 연속한 구간별로 원래 길이의 범위를 구하고 가능한 괄호 잔량을 구간으로 유지한다.

화면에서 같은 괄호가 연속한 최대 구간의 길이를 r이라 하자. 키 한 번이 한 글자나 두 글자가 되므로 원래 누른 횟수는 $a = \lceil \frac{r}{2} \rceil$ 이상 $b = r$ 이하의 모든 정수가 가능하다. 서로 다른 괄호 사이의 경계는 중복 입력으로 바뀌지 않는다.

접두사를 복구했을 때 아직 닫히지 않은 여는 괄호 수를 잔량이라고 하자. 잔량은 항상 음이 아니어야 하며 마지막에는 0이어야 한다.

가능한 잔량 전체를 정수 구간 $[L, H]$로 관리한다. 처음에는 $[0, 0]$이다. 이번 연속 구간의 원래 길이 범위가 $[a, b]$이면 다음처럼 갱신한다.

- 여는 괄호 구간이면 $[L + a, H + b]$로 바뀐다.
- 닫는 괄호 구간이면 $[\max(0, L - b), H - a]$로 바뀐다.

두 정수 구간의 합과 차에는 중간에 빈 정수가 없으므로 가능한 잔량도 계속 구간이다. 닫는 구간에서는 잔량이 줄어들기만 하므로 끝의 잔량이 음수가 아니면 그 구간 내부에서도 올바르다.

갱신 후 $L > H$라면 가능한 복구가 없어졌으므로 즉시 `NO`이다. 모든 구간을 처리한 뒤 $L = 0$이면 잔량 0을 만들 수 있으므로 `YES`, 아니면 `NO`이다. 관측된 문자열의 길이가 홀수여도 복구는 가능할 수 있다.

복잡도: 시간 $O(N)$, 문자열 저장을 제외한 추가 공간 $O(1)$이다.

D. 스위스 치즈 모델 (Easy)

핵심: 구멍 하나를 정점으로 보고 서로 이동할 수 있는 구멍을 간선으로 연결한다.

한 구멍의 내부는 모두 이어져 있다. 같은 치즈에 있는 두 구멍이나 이웃한 두 치즈에 있는 두 구멍은 단면 원판이 겹치거나 접하면 서로 이동할 수 있다. 따라서 이 관계를 무방향 그래프로 표현한다.

두 원판의 중심과 반지름이 각각 (x_1, y_1, r_1)과 (x_2, y_2, r_2)이면 연결 조건은 다음과 같다.

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 \le (r_1 + r_2)^2.$$

경계 위의 이동도 허용되므로 등호를 포함한다. 원판 하나가 다른 원판 안에 들어 있는 경우도 연결되어 있다. 같은 치즈도 이웃한 치즈도 아닌 구멍끼리는 간선을 만들지 않는다.

첫 치즈의 모든 구멍을 시작점으로 넣어 BFS를 한다. 마지막 치즈의 구멍 중 하나라도 도달 가능하면 사고가 통과할 수 있다. 구멍 안에서 필요한 교차점까지 이동할 수 있으므로 그래프 경로는 실제 이동 경로로도 이어진다. 반대로 실제 이동에서 구멍을 바꾸는 순간마다 위 연결 조건을 만족하므로 그래프 탐색으로 충분하다.

앞쪽 치즈로 돌아가는 이동도 허용하므로 간선은 무방향이다. 같은 치즈 안에서 여러 구멍을 거칠 수도 있어 이웃한 치즈 사이의 연결만 검사하면 안 된다.

유니언 파인드로도 풀 수 있다. 이동 가능한 구멍 쌍을 같은 집합으로 합친다. 가상의 시작점과 끝점을 만들고 각각 첫 치즈와 마지막 치즈의 모든 구멍과 합친 뒤 두 가상 정점이 같은 집합인지 확인하면 된다. 간선을 따로 저장하지 않으므로 추가 공간은 $O(M)$이다.

출력 주의: 사고가 통과할 수 있으면 안전하지 않으므로 `NO`, 끝까지 도달하지 못하면 안전하므로 `YES`이다.

복잡도: 전체 구멍 수를 M이라 할 때 모든 쌍을 검사하고 BFS를 하면 시간과 공간 모두 $O(M^2)$이다.

E. 그래프 눌러담기

핵심: 기존 사이클 내부를 제외하면 정점 쌍마다 한 방향의 간선을 채울 수 있다.

함수형 그래프의 각 약한 연결 성분에는 유향 사이클이 하나씩 있다. 사이클을 하나의 정점으로 압축하면 사이클이 없는 유향 그래프가 된다.

같은 기존 사이클의 두 정점 사이에는 이미 양방향으로 경로가 있다. 따라서 그 안에 없던 간선을 추가하면 새로운 사이클이 생긴다. 반면 같은 사이클에 속하지 않는 두 정점 사이에는 두 방향을 모두 둘 수 없다. 새 2-사이클이 생기기 때문이다.

이 상한은 실제로 달성할 수 있다. 압축한 그래프를 위상 정렬한 뒤 서로 다른 압축 정점 사이의 모든 정점 쌍에 대해 앞에서 뒤로 향하는 간선을 채운다. 기존 간선의 방향을 유지하며 압축 그래프에 사이클이 생기지 않으므로 새로운 사이클도 생기지 않는다.

기존 사이클들의 크기를 $c_1, c_2, \ldots$라 하자. 전체 서로 다른 정점 쌍은 $\frac{N(N-1)}{2}$개이며 같은 사이클 내부의 쌍은 추가에 사용할 수 없다. 또 사이클 밖의 기존 간선 $N - \sum_j c_j$개는 이미 사용할 수 있는 쌍을 차지하고 있다. 따라서 정답은 다음과 같다.

$$\frac{N(N-1)}{2} - \sum_j \frac{c_j(c_j-1)}{2} - (N - \sum_j c_j).$$

사이클 크기는 진입차수가 0인 정점을 큐에서 꺼내 제거하는 방법으로 구한다. 제거되지 않은 정점은 모두 사이클 위에 있으므로 아직 방문하지 않은 정점에서 간선을 따라가며 크기를 센다. 자기 루프는 크기 1인 사이클로 처리한다.

복잡도: 시간과 공간 모두 $O(N)$이다. 정답은 32비트를 넘을 수 있으므로 곱셈과 합산에 64비트 정수를 사용한다.

F. 스위스 치즈 모델 (Hard)

핵심: 모든 치즈의 구멍 합집합에 동시에 속하는 평면 좌표가 있는지 확인한다.

직선 경로는 모든 치즈에서 같은 좌표를 지나므로 Easy처럼 구멍이 이어지는 것만으로는 충분하지 않다. 대신 **모든 원의 중심과 두 원의 둘레 교점**을 후보로 검사한다.

이 후보들만 보면 되는 이유는 다음과 같다. 통과 가능한 점이 있다면 각 치즈에서 그 점을 포함하는 원판 하나씩을 고르자. 이 원판들의 교집합은 비어 있지 않은 닫힌 볼록 집합이다.

- 교집합이 한 원판 전체라면 그 원의 중심이 모든 선택 원판에 포함된다.
- 그렇지 않으면 교집합의 경계를 이루는 원호가 바뀌는 점이 있다. 이 점은 두 원의 둘레 교점이며 모든 선택 원판에 포함된다. 교집합이 한 점뿐인 경우에도 접점이 후보가 된다.

따라서 답이 존재하면 중심이나 교점 중에도 답이 있다. 후보점 하나에 대해 **치즈마다 적어도 한 구멍에 포함되는지** 검사한다. 모든 구멍에 포함될 필요는 없다.

교점은 다음 식으로 구할 수 있다. 중심을 c_1, c_2, 중심 거리의 제곱을 d^2, 거리를 d, 두 중심을 잇는 단위 벡터를 u, 이에 수직인 단위 벡터를 v라 하자.

$$a = \frac{r_1^2 - r_2^2 + d^2}{2d}, \qquad s = r_1 + r_2, \quad t = r_1 - r_2,$$
$$h = \sqrt{\max\left(0, \frac{(s^2 - d^2)(d^2 - t^2)}{4d^2}\right)},$$
$$p_{\pm} = c_1 + au \pm hv.$$

교점은 $|r_1 - r_2| \le d \le r_1 + r_2$일 때만 실수로 계산한다. 중심이 같으면 건너뛰며 접점도 포함한다. 위 높이 식은 접점 근처의 빨셈 오차를 줄인다. 포함 판정에는 지문의 반지름 여유 조건보다 작은 거리 허용오차를 사용한다.

복잡도: 전체 구멍 수가 M이면 후보 $O(M^2)$개를 각각 $O(M)$에 검사하여 시간 $O(M^3)$이다. 후보를 저장하면 공간 $O(M^2)$이며 즉시 검사하면 $O(M)$이다. 통과 후보가 있으면 `NO`, 없으면 `YES`를 출력한다.

G. 패셔니스타

핵심: 최근 6일에 입은 옷의 실제 이름 대신 같은 옷을 입은 날짜들의 관계만 기억한다.

앞으로의 조건은 최근 6일만 참조한다. 그러나 금지된 날짜가 여러 개여도 같은 옷을 입었다면 금지되는 옷은 한 벌이므로 단순히 $7-k_i$를 곱할 수는 없다.

최근 옷들을 처음 나타난 순서대로 0, 1, 2 등으로 바꾸어 상태로 삼는다. 예를 들어 `[5,2,5,7]`은 `[0,1,0,2]`로 표현한다. 이런 같고 다름의 패턴은 길이가 6일 때 203개이고 길이가 0부터 6까지인 상태를 모두 합쳐도 279개뿐이다.

`dp[state]`를 그 패턴을 갖는 실제 옷차림 계획 수라고 하자. 빈 상태의 값을 1로 시작한다. 오늘의 금지 날짜들이 가리키는 라벨을 모은 뒤 다음 두 종류의 전이를 한다.

- 최근 6일에 등장한 라벨 중 금지되지 않은 라벨을 고른다. 해당 옷은 한 벌이므로 경우의 수 배수는 1이다.
- 최근 6일에 없던 옷을 고른다. 상태에 서로 다른 라벨이 u개라면 실제 선택은 $7-u$가지이고 모두 같은 새 라벨 상태로 이어진다. 따라서 경우의 수에 $7-u$를 곱한다.

고른 라벨을 뒤에 붙이고 길이가 6을 넘으면 가장 오래된 값을 지운다. 그 뒤 **다시 처음 등장한 순서로 라벨을 붙여** 상태를 정규화한다. 같은 다음 상태로 가는 경우의 수는 더하고 매번 998244353으로 나눈다.

과거에 입었더라도 최근 6일에 없는 옷은 두 번째 전이에 포함된다. 실제 옷 이름을 바꾸어도 조건은 같으므로 이 DP는 각 실제 선택을 빠짐없이 센다. 마지막 상태들의 값을 모두 더하면 정답이다.

복잡도: 금지 거리 집합을 6비트로 표현하고 각 상태의 전이를 미리 계산한다. 날짜마다 상태당 최대 7개 선택을 처리하므로 시간 $O(N \cdot 279 \cdot 7)$이다. 전이표와 입력을 저장하면 공간 $O(279 \cdot 64 \cdot 7 + N)$이다.