**문제 A: 달구의 카드게임**

첫번째 시도 : 숫자 카드의 모든 조합을 만들고, 카드가 들어올 때마다 업데이트

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* 시간초과(O(2^n) & O(n^2)

두번째 시도 : binary search를 이용해 차이를 구하고, 카드에 값이 있는지 확인

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* 시간초과(O(nlogn))

세번째 시도 : 누적합을 이용한 reachable 요소 설정

작은 요소들부터 하나씩 추가해나가면 대략적인 coverage를 알 수 있다고 판단.

만들어둔 test case(K=10, n=[1,3,5]에서 correctness를 보장하지 못했으나 time complexity를 O(n)으로 줄였다는 점이 출제 의도에 부합한다고 판단

**문제 B: 달구의 음식점**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Kruskal algorithm : greedy algorithm을 이용하여 minimum spanning tree를 찾는 방법

테이블 사이의 거리가 weight의 역할을 하고, 테이블을 node의 역할을 한다고 생각할 때, 모든 테이블에 음식을 배달할 수 있도록 하기 위해서는 spanning tree를 만들어야 한다고 판단

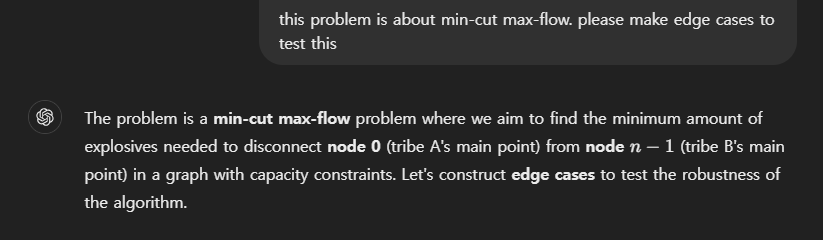
로봇의 메모리가 한정되어있다는 점은 spanning tree 중에서도 minimum이 되어야 함을 의미한다고 판단

따라서 받은 입력을 통해 그래프를 만들고, Kruskal algorithm을 이용하도록 설계

**문제 C: 부족 전쟁**

부족 A의 입장에서 적들이 가장 많이 오는 경로를 끊어야 한다. 적들이 건너오는 다리의 길이를 weight로, 거점을 node로 생각하였다.  
문제에서 요구하는 바는 ‘최소한의 폭약’ 즉, 거점A와 거점B 사이의 연결이 끊어질 때까지 폭약을 사용할 때 minimum sum of weight라고 생각하였다.(min cut)

이는 Max-flow Min-cut theorem에 의해 max flow와 동일하므로 Ford-Fulkerson algorithm을 사용하는 방향으로 구성하였다.



GPT4o 모델을 통해 edge case를 생성하였으며, 주요 포인트는 K 이상의 다리 길이를 갖는 부분을 min cut에서 제외하는 것이었다. Min cut은 node를 서로 다른 집합으로 나누기 위해서 잘라야 하는 edge들 중 가장 weight 값의 합이 작은 것이다. 따라서 어느 하나의 edge라도 min cut에서 제외되는 순간 두 집합이 연결되기에 -1이 return 되도록 하였다.

참조 사이트 : [백준] 간선 끊어가기 2

**문제 D: 달구의 장애물**

N\*N 좌표를 그래프라고 생각하고, 선을 edge들이라고 생각하였다. 그리고 각각의 weight는 모두 1로 동일하다고 설정하였다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

C의 알고리즘을 그대로 옮겨온 경우, test case에서 위와 같이 점 주위 모든 edge를 잘라버리는 방식으로 작동하였다.