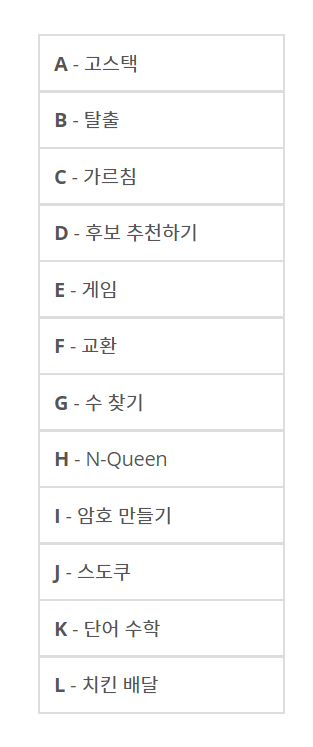
1.알고리즘 기초



탈출을 다시 풀어봄 🡪 역시나 처음 푸는 느낌.. // 아무튼 여기서 내 사고 과정의 포인트는 최단거리니까 비버는 현 상태에 크기만큼만 bfs 돌려줌 // 그리고 물 먼저 해버려서 못 가게함// 생각해 보니 check 배열을 공유했어도 괜찮았을거 같음 이미 물이 올 곳을 체크해주면 비버는 거기를 못가니까// 아 그리고 CNT 위치 주의//

가르침도 re 🡪 시간 오래걸렸던거 flag 위치 리셋을 안시켜줌… 그리고 auto 쓰는거 보다 그냥 문자열 배열로 자리 찾는게 빠름//

후보추천하기🡪 솔직히 추천수를 따로 빼는걸 생각을 못함 // 근데 등장 순서도 독립적으로 뺄 수 있음// 중요한건 두 자료구조의 연결고리를 만드는게 중요// 결국 비교해야 되니까//연결고리는 후보 번호임// 그리고 그냥 이중 포문으로 계속 보는거임 // 하 이거 조건 잘못봤음// 항상 문제 잘 보기//

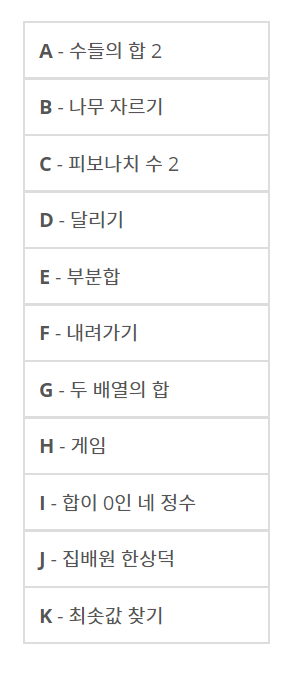
게임 re🡪 아이거 dp인데 생각 또 못함 ㅅㅂ/// ch 사용해서 전에꺼 다녀왔으면 무한 루프임 그래서 탈출 dp는 x,y위치 일 때 몇번 게임 했냐// 완탐이랑 dp차이 생각해서 코드 짜라

교환 re 🡪 bfs 응용// 구현 문제임// bfs의 특징을 잘 이해하고, 턴 구별하는 것 중요// 이건 다시 풀어봐야할 듯

n-queen🡪 이건 왜 항상 못 풀까/// 이거 일반 버전 있는데 좀 오바인듯//

단어수학 re🡪 벡터 크기 정해 놓은 것 꼭 0부터 시작을 의미 하지는 않음// 26이라고 사이즈는 만들어 놓고 다른 값에 넣어 줄 수가 있음 근데 굳이 ../// 수학처럼 푸는 법 다시 생각//

2.시간 복잡도



수들의합 re🡪 투포인터 문제//

달리기 re🡪 다시품 펜윅으로 // 펜윅이 1부터 인덱스가 시작하는데 계속 0으로 해서 문제가 있었음//

최솟값 찾기re 🡪 데크 응용을 잘 생각해 보자(슬라이딩 윈도우)// 우선 순위 큐를 사용해서 풀이 할 방법은

내려가기 re 🡪 dp 최대 최소 나눠서//

두 배열의 합 re🡺 이분탐색// 누적합해서 구함 왜냐? 어차피 쌍의 수니까 모든 경우의 수를 이분탐색 하는거임 lower upper 써서 개수 구함//

게임 re

합이 0인 네 정수 re🡪 두 배열의 합이랑 비슷// lower-bound upper\_bound 사용해서 개수 구하는 구조./

\*집배원 한상덕 re🡪 set 사용해서 투 포인터 응용 하는 법 연습//

3.자료구조



트리 인가? Re 🡪 dfs로 푸는데 자식들도 체크 해줘야 되고 set 사용해서 root일 때 조건도 잘 봐야됨 예외 처리들이 좀 있음//

보석도덕 re 🡪 전에는 mutiset 사용해서 풀었는데 이러면 훨씬 쉬움 그냥 가격 높은 순으로 되어있는 우선순위 큐에서 하나씩 빼서 멀티셋에서 그에 맞는 무게 있으면 값 추가해줌 /// 근데 그냥 우선순위 큐랑 배열 사용해서 풀려면// 보석이랑 가방을 오름차순으로 정렬하고 가방 무게에 맞는 보석을 찾아서 우선순위 큐에 넣고 가장 높은 값을 구해서 더하고 계속 반복//

사탕상자 re 🡪 개수 세는거 위치 찾기// 순위 라는 것이 결국은 맛마다 개수들이 여러 개 있으니까 그에 맞게 세어줘야됨//update 해줄 때 맛이 index으로 트리의 개수에 추가함// 따라서 순위에 맞는 맛을 찾을 때 는 그 맛의 개수들을 비교해서 왼쪽 오른쪽으로 나눠서 출력함 현 인덱스 트리 값보다 작으면 그 안쪽이니까 왼쪽 아니면 바깥쪽// 그리고 왜 maxn 값까지 하는지 update 할때/ 🡪 사탕맛은 다 있을 수도 있으니까//

개똥벌레 re🡪

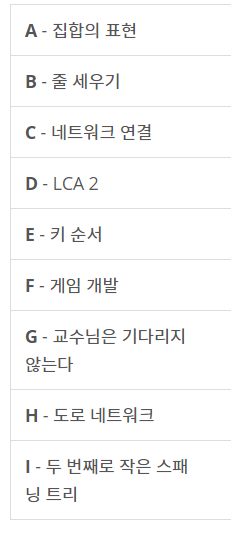
4. 정수론



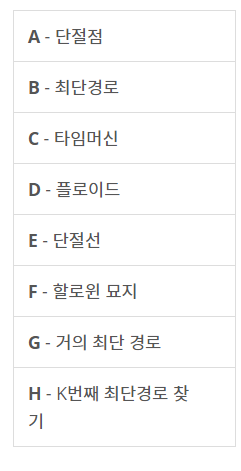
5. 조합론



6.그래프1



7.그래프2



8.동적계획법1



9.동적계획법2

