

코드포스 1주차 문제 풀이

문제 난이도

	문제 번호	문제 이름	난이도
A	14656	조교는 새디스트야!!!	B3
B	10040	투표	B2
C	3896	소수 사이 수열	S1
D	20417	역전의 제왕(Normal)	S1
E	27498	연애 혁명	G3
F	20327	배열 돌리기 6	G2
G	26615	다오의 행사 계획하기	P4
H	2726	삼각 N-Queen	P4

A. 조교는 새디스트야!!!

구현

A. 조교는 새디스트야!!!

순서대로 서있지 않은 학생들이 몇명인지 세는 문제입니다.

idx = 1이라고 놓고 배열을 돌면서 학생들의 번호와 idx가 맞는지 판단하고
아니면 정답 += 1 입니다.

시간복잡도 : $O(N)$

B. 투표

구현, 시뮬레이션

B. 투표

N개의 문제가 있고, M명의 심사위원이 있습니다.

각 심사위원의 심사 기준보다 비용이 적으면서 가장 재미있는 경기를 찾아야 합니다.

재미있는 경기 순서대로 입력이 주어지므로 위에서부터 탐색하다가 A_i 가 B_j 보다 작거나 같으면 그 때의 경기가 정답입니다.

시간 복잡도 : $O(NM)$

C. 소수 사이 순열

수학, 정수론, 이분탐색, 소수 판정, 에라토스테네스의 체

C. 소수 사이 수열

수가 주어졌을 때, 소수면 0을 출력하고 합성수면 앞뒤의 소수의 간격을 출력하는 문제입니다.

에라토스테네스의 체를 입력의 최대치인 1299709 까지 적용시키고 이분탐색하여 앞뒤의 소수를 알아내서 간격을 출력합니다.

소수의 간격은 크지 않기 때문에, 그냥 앞뒤로 소수가 나올때 까지 탐색하여 출력하는 방법도 있습니다.

시간 복잡도 : 에라토스테네스의 체 ->	$O(N \lg \lg N)$
앞뒤로 나올 때 까지 탐색 ->	$O(\sqrt{N})$

D. 역전의 제왕 (NORMAL)

구현, 시뮬레이션

D. 역전의 제왕 (Normal)

순위를 산정하는 우선순위는 다음과 같습니다.

1. 해결한 문제의 수가 많은 순서
2. 패널티 총합이 적은 순서
3. 마지막 정답 제출 시간이 빠른 순서 (제출된 인덱스 순서)

시간 순서대로 입력이 주어지기 때문에 03:00 까지 순위를 산정하고 그 이후로는 제일 꼴등부터 제출을 적용시킵니다.

이 때, 제출한 문제가 여러개인 경우, 문제의 번호가 가장 빠른 순서로 제출이 공개되고 순위 변동이 있으면 올라간 순위만큼 역전 포인트를 얻습니다.

D. 역전의 제왕 (Normal)

역전 포인트가 동일한 참가자가 있는 경우엔, 최종 순위가 가장 높은 참가자가 정답입니다.

이 문제 구현을 위해 구조체나 클래스를 사용하여 정렬하는 방법, 우선순위 큐를 사용하는 방법도 있습니다.

시간복잡도 : $O(M \lg N)$

E. 연애 혁명

그래프 이론, 최소 스패닝 트리

E. 연애 혁명

학생들 사이의 복잡한 사랑 관계를 단순화하여 K 각 관계($K \geq 3$)가 발생하지 않도록 해야 합니다. 즉, 사이클이 있으면 안되고 각각을 트리 구조로 만들어야 합니다.

이미 성사된 사랑을 제외하고 포기하는 애정도의 합이 최소가 되게 해야합니다.

Union-Find 를 이용한 최소 스패닝 트리를 구현해볼 수 있습니다.

이 때 애정도의 합이 최소가 되어야 하기 때문에 아직 성사되지 않은 애정도를 정렬하는 과정이 필요합니다.

E. 연애 혁명

성사된 것은 Union-Find 자료구조로 이어주어야 합니다. 성사되지 않은 것은 애정도 기준 오름차순으로 정렬하여 이어지지 않았다면 이어주고, 이미 이어져 있다면 그 때의 애정도를 전부 합한 것이 답입니다.

14주차에 Union-Find와 스패닝 트리를 다룰 것이기 때문에 이해되지 않아도 괜찮습니다.

시간복잡도 : $O(M \lg M)$

F. 배열 돌리기 6

구현, 시뮬레이션

F. 배열 돌리기 6

ℓ 과 연산 종류가 주어지면 그에 맞는 구현을 해야 합니다.

1, 2, 3, 4는 부분 배열 내의 연산이고 5, 6, 7, 8은 부분 배열을 하나로 가정한 전체적인 연산입니다.

구현 팁

1. 부분 배열 추출 및 연산

ℓ 에 따라 배열을 추출하고 연산합니다.

2. 배열 조작을 함수로 구현

회전 연산, swap 연산을 줄 단위로 구현할 수 있습니다.

시간복잡도 : $O(R \times 2^{2N})$

G. 다오의 행사 계획하기

트리, 누적 합, 최소 공통 조상

G. 다오의 행사 계획하기

다오가 각 날짜마다 행사장에 올 사람들의 수를 알고 싶어 합니다.

행사장의 임의의 두 칸을 잇는 경로는 정확히 1개 있습니다. 즉, 트리구조입니다.

T가 주어지고 시작 일, 마감 일, 시작 칸, 도착 칸, 오는 사람의 수가 주어집니다.

시작칸 ~ 도착칸의 경로에서 몇 칸이 포함되는지 구하고 오는 사람의 수를 곱하면 그 날 오는 사람의 수를 구할 수 있습니다.

이를 dfs 혹은 bfs로 구할 수 있으나, 이는 $O(KNM)$ 이므로 TLE 입니다.

G. 다오의 행사 계획하기

우리는 이를 LCA(최소 공통 조상) 알고리즘을 이용하여 $O(K \lg(NM))$ 에 구할 수 있습니다.

구한 값을 in배열에는 시작일 인덱스에, out배열에는 마감일 인덱스에 누적하여 더합니다.

1부터 T까지 순회하면서 in배열에 값이 있으면(사람이 왔으면) 그 만큼 더하고, out배열에 값이 있으면(사람이 나갔으면) 그 만큼 빼서 출력하면 정답입니다.

이 알고리즘은 코딩 테스트에 나오는 유형은 아니라 이해가 안가도 괜찮습니다.

시간복잡도 : $O(NM + K + T)$

H. 삼각 N-QUEEN

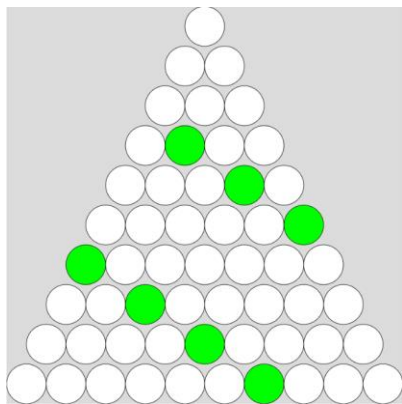
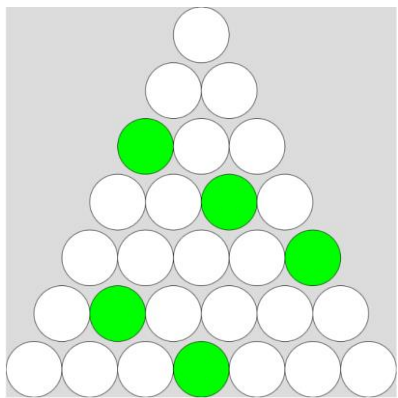
수학, 해 구성하기

H. 삼각 N-Queen

우리는 한 변의 길이가 N인 삼각형에 서로 공격할 수 없는 위치에 퀸을 최대한 많이 놓아야 합니다.

문제에서 주어진 바로는, 항상 $\text{floor}(\frac{2N+1}{3})$ 개의 퀸의 배치가 가능합니다.

N이 6인 경우까지는 할만한데 N이 7인 경우 5개 배치를 해야 합니다.

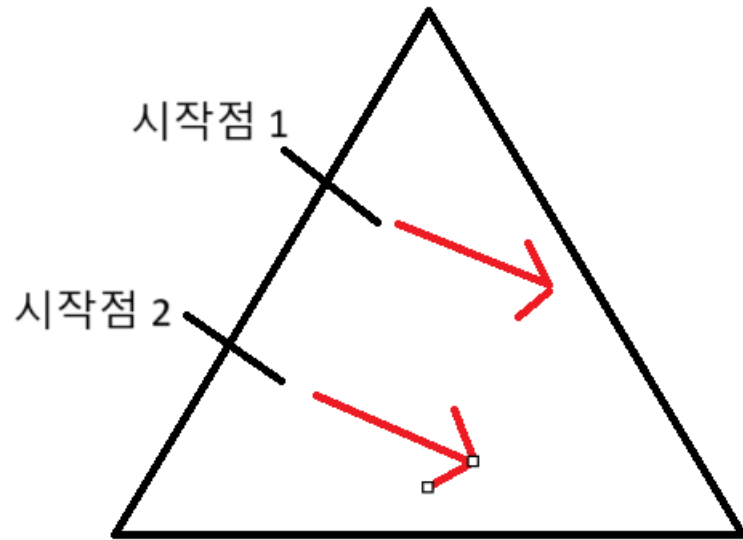


N이 7인 경우와 N이 10인 경우입니다.

감이 오시나요?

H. 삼각 N-Queen

변의 3분의 1지점을 기점으로 해서 한 줄씩 내려가면서 배치하는 것입니다.
시작점1에서 처음 인덱스가 0이라면, 시작점2의 처음 인덱스는 1이고



시작점1에서 처음 인덱스가 1이면, 시작점 2의
처음 인덱스는 0입니다.

이를 이용하여 한 줄 씩 내려가면서 출력하면
정답입니다.

시각화 도구 : <https://editor.p5js.org/unta1337/sketches/Kzt53Qaee>

시간 복잡도 : $O(TN)$