Semi – Game cup 풀이

Made by stonejjun03

권고 사항

- 문제 중에 아이디어가 거의 모든 부분을 차지하는 문제가 꽤 있으니 충분한 시간을 고민한 후에 그 문제에 대한 풀이 부분을 봐주시면 감사하겠습니다...

A.왕들의 외나무 다리 돌 게임

stonejjun03

A.왕들의 외나무 다리 돌 게임 (1)

- 기본적으로 스프러그 – 그런디 정리나 그런디 수를 알고 있는 것이 좋다.
- 일단 N개의 돌무더기가 있고 각 돌 무더기에는 Ai개의 돌이 있으며 자신의 차례에 어떤 한 돌무더기에서 원하는 만큼 가져가는 게임을 생각해보자.
- 이 게임은 스프러그 – 그런디 정리에 의해 Ai들을 다 XOR 한 값이 0이면 후공 승, 그렇지 않으면 선공이 승리한다.

A.왕들의 외나무 다리 돌 게임 (2)

- 그러면 이 게임에 적용시켜보자.

- 핵심적인 부분은 나와 상대 돌사이의 남은 칸이다.
- 내가 행동을 함으로써 필승 상황이 만들어졌으면, 상대가 뒤로 간만큼 나는 앞으로 감으로써 똑같은 상황을 연출할 수 있다.

- 즉, 뒤로 가게 되는 경우가 존재하는 것은 승자가 정해져 있는 것에 아무런 영향을 끼치지 못한다는 것이다.

- 따라서 아까의 무더기 돌 게임과 같은 형식이고, Ai-2 들을 모두 XOR 한 값이 0인지를 확인해 주면 된다.

B. 돌 술래잡기 게임

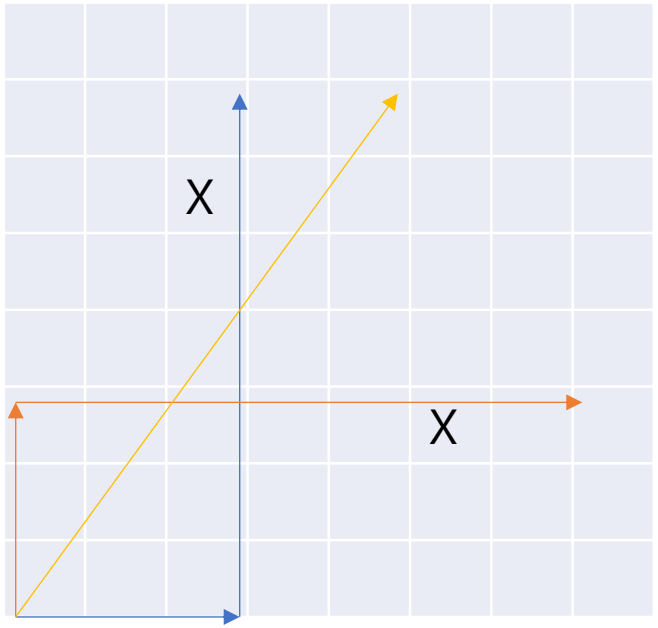
stonejjun03

B. 돌 술래잡기 게임 (1)

- 0 번째로 x=y 상에 돌이 있으면 무조건 잡는다.
- 그 후로, 일단 정말 중요하면서도 핵심적인 첫번째 관찰이 필요하다.
- ‘검은 돌이 잡을 수 있는 상황이다.’ 는 ‘전체 검은 돌의 집합 중, 특정 2개의 검은 돌만 있어도 잡을 수 있다.’와 동치라는 것이다.
- 따라서 돌 두 개가 있을 때 그 돌에 대해 흰 돌이 탈출할 수 있는지 판단, 다시 말해 흰 돌을 잡을 수 있는 돌 한 쌍이 있는지 판단할 수 있으면 된다.

B. 돌 술래잡기 게임 (2)

- 두 개의 검은 돌에 대한 흰 돌의 최적 경우를 생각해보자. 기본적으로 생각해야할 부분은 어떤 순간에도 흰 돌과 검은돌의 x-y가 같게 되는 순간이 있으면 안된다. 또한 당연히 x=y 기준으로 다른쪽의 두 돌이다.

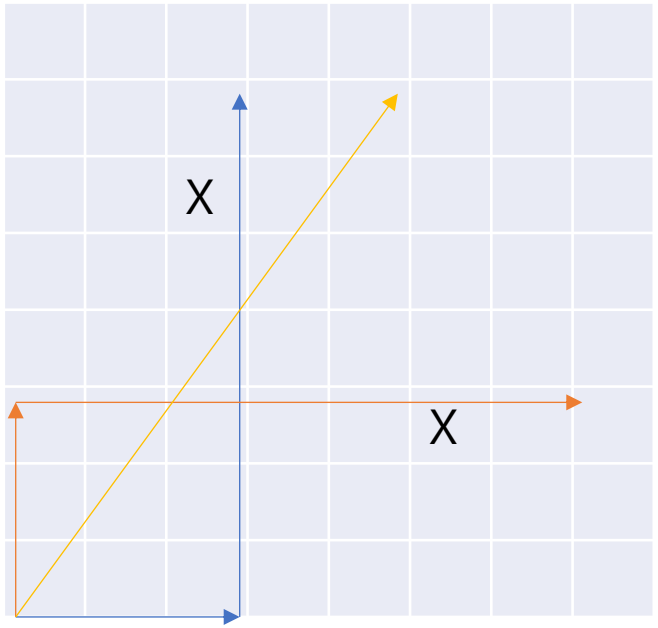


일단 둘 중 하나의 돌을 무력화 시키고 가는 방법이 존재한다. 이때 다른 한 돌에 대해서 x-y가 같지 않으면 무조건 살 수 있다.

혹은 두 돌 사이로 x-y가 같지 않게 계속 조정하면서 빠져나갈 수 있다.

B. 돌 술래잡기 게임 (3)

- 두 개의 검은 돌에 대한 흰 돌의 최적 경우를 생각해보자. 기본적으로 생각해야할 부분은 어떤 순간에도 흰 돌과 검은돌의 x-y가 같게 되는 순간이 있으면 안된다.



이를 수직선상에 놓고 각 돌을 x-y 위치에 놓은 뒤, 검은돌이 먼저 좁혀서 토끼몰이가 될 지, 그 전에 한 검은돌의 x+y(수명)이 다 될지로 생각해서 풀어도 된다.

B. 돌 술래잡기 게임 (4)

- 두 경우모두 흰 돌의 가능한 최선의 전략이 3가지고 검은돌 입장에서는 그 3가지 전략을 모두 막을 수 있는 두 돌을 찾으면 된다.
- 그 3가지 조건은 돌의 x,y좌표에 대한 3가지 식으로 나오게 되고, 간단히 부등호 하나 있는 식이 나온다.
- 따라서 3개의 인자가 모두 같거나 작은 쌍을 찾아야 한다.
- 이는 세그트리를 이용하여 O(NlgN)에 해결할 수 있다.

C. 경비병 세우기 게임

stonejjun03

C. 경비병 세우기 게임 (1)

- 어떤 판의 상황에서 판의 내부에 완전히 포함되는 k*k의 정사각형 중 경비병이 하나도 설치되지 않은 정사각형들의 집합을 S라고 하자.
- S들의 교집합이 한 칸이라도 있으면 바로 놓음으로써 게임을 끝낼 수 있으므로 상대에게는 그런 상황을 주어서는 안된다.
- 현재 S들의 교집합이 없는 상황이라고 생각해보자.
- 이때 우리는 서로 겹치지 않는 두 정사각형을 무조건 고를 수 있다.

C. 경비병 세우기 게임 (2)

- Pf. 수직선에서 구간들이 있을 때 전체 구간들에 대해서 교집합이 없으면, 서로 겹치지 않는 두 구간이 반드시 존재하고 이를 2차원으로 확장시켜 생각하면 이해는 된다.

- 따라서 그 두 정사각형을 뺀 나머지 칸에 놓기만 하면 내가 이기지 않는한, 상대는 절대로 이길 수 없게 된다. (필승법)

- 판에서 그러한 칸의 개수는 nm-2*k*k 이므로 nm이 홀수이면 선공이 항상 그렇게 놓을 수 있고, nm이 짝수이면 후공이 항상 그렇게 놓을 수 있음이 보장된다.

- 이와 별개로 선공이 첫 턴에 끝내는 것은 따로 고려해주자.

D. 숫자 카드 제거 게임

stonejjun03

D. 숫자카드 제거게임

- 이 문제도 기본적으로 스프러그 – 그런디 정리나 그런디 수를 알고 있는 것이 좋다.

- 1부터 N까지 연속되어 있는 상태에서 한 턴이 지나면 나올 수 있는 경우의 수는 3가지다.

1. 연속된 길이가 N-2가 된다.
2. 연속된 길이가 N-3이 된다
3. 길이의 합이 N-3인 두 개의 파트로 쪼개진다.

따라서 이 값에 대해서 각각의 그런디 수를 구해서 mex를 취해주면 현재 판의 그런디 수를 알 수 있다.

D. 숫자카드 제거게임

- 하지만 그렇게 하면 O(N^2)이기 때문에 문제를 해결할 수 없다.

- 하지만 후공이 이기는 경우를 출력해보면 34를 주기로 규칙이 있다는 것을 확인할 수 있다.
- 하지만 악랄하게도 예외가 있기 때문에 조금의 예외 처리를 해주어야 한다.

- 이는 N+1개의 칸에 도미노를 놓는 게임과 동치로, https://oeis.org/A215721?fbclid=IwAR33UIpCKHPQ7dAK_O6JmZbG-QTap0R0z3GbUzHmkQKVM6ndTuT3O_iNi4A 에서 그 수열을 확인 할 수 있다.

E. 평면그래프와 게임

Written by ahgus89 / Made by TAMREF

문제 설명

- 평면그래프가 주어진다.
- 두 정점을 잇는 간선이 있다면 끊는 쿼리가 있다.
- 두 정점이 같은 컴포넌트에 속하는지 묻는 쿼리가 있다.
- **온라인이다.**

풀이

- 처음에 Union-find나 BFS, DFS 등으로 정점을 컴포넌트별로 분리할 수 있다.
- u, v를 잇는 간선을 끊는 쿼리를 처리해보자.
- 우선 u, v의 연결을 해제한다.
- 컴포넌트가 분리된다면, 둘 중 한 정점을 골라 연결된 모든 정점에 새로운 컴포넌트 번호를 붙여주면 된다.
- u, v가 포함된 컴포넌트 중 더 작은 쪽에만 새로운 번호를 붙여주면, (smaller from larger?) 이 과정은 O(NlogN)에 된다.

풀이

- 컴포넌트가 분리되는지는 어떻게 확인할까?
- 주어진 그래프가 평면 그래프이므로, $v - e + f = c$가 성립한다. (이때 c는 컴포넌트 개수이다.)
- v는 변하지 않고, e는 간선을 끊을 때마다 하나씩 감소한다.
- 따라서 간선을 끊었는데도 f가 감소하지 않은 경우가 컴포넌트가 분리된 경우이다.
- 이는 주어진 그래프의 dual에서 Union-find를 이용해 해결할 수 있다.
- 따라서 주어진 문제를 O(NlogN)에 쉽게 해결할 수 있다.

F. 고인물의 새로운 리듬게임

stonejjun03

F. 고인물의 새로운 리듬게임

- N개중 K개 이하를 고르는 데 최댓값. DP로 이 문제를 해결 할 수 있다.

- DP[i][j]를 i번째 노트까지 중에서 j개의 노트를 쳤을 때 얻을 수 있는 최대 점수라고 하자.

- 이에 대해서 DP식을 세워보면

$$DP[i][j] = \max_{k<i}(DP[i-k-1][j-k] + F(i-k+1, i)) + P$$

이때 F(x,y)는 x번째 노트부터 y번째 노트 까지만 쳤을 때 얻을 수 있는 점수이다.

F. 고인물의 새로운 리듬게임

- 하지만 DP는 2차원이고 한 칸당 O(N)이므로 총 O(N^3)의 시간복잡도를 가지게 된다.
- 단조성에서 눈치를 챌 수 있지만 DP opt를 써야한다.

- 같은 i-j 묶음들에 대하여 i가 증가할수록 최댓값을 가지는 i-k는 같거나 증가하기 때문에 DNC opt를 쓸 수 있는 조건이 나오게 되었다.

- 따라서 DNC opt를 사용해 주게 되면 O(N^2lgN)에 해결할 수 있다.

- 이 밖에 Lichao tree, monotone queue opt등을 사용하는 풀이도 존재한다.

G. 지역 꾸미기 게임

Blackkin26

G 지역 꾸미기 게임

- 입ps로는 가능한 쿼리를 실제로 구현해야 하는 문제이다.
- 일반적으로 2d range update/query는 당연히 못한다.
- update를 공유하는 각 조각에 대해서 가장 왼쪽 위에만 update를 해주면 점 업데이트 구간 쿼리로 바뀐다.
- 이것으로 2번과 3번 쿼리는 평범한 2d segment tree로 처리할 수 있게 되지만…
- 1번 쿼리를 처리하기 위해서는 쿼리 한번 당 최대 O(N)개의 정점을 추가해야 한다.

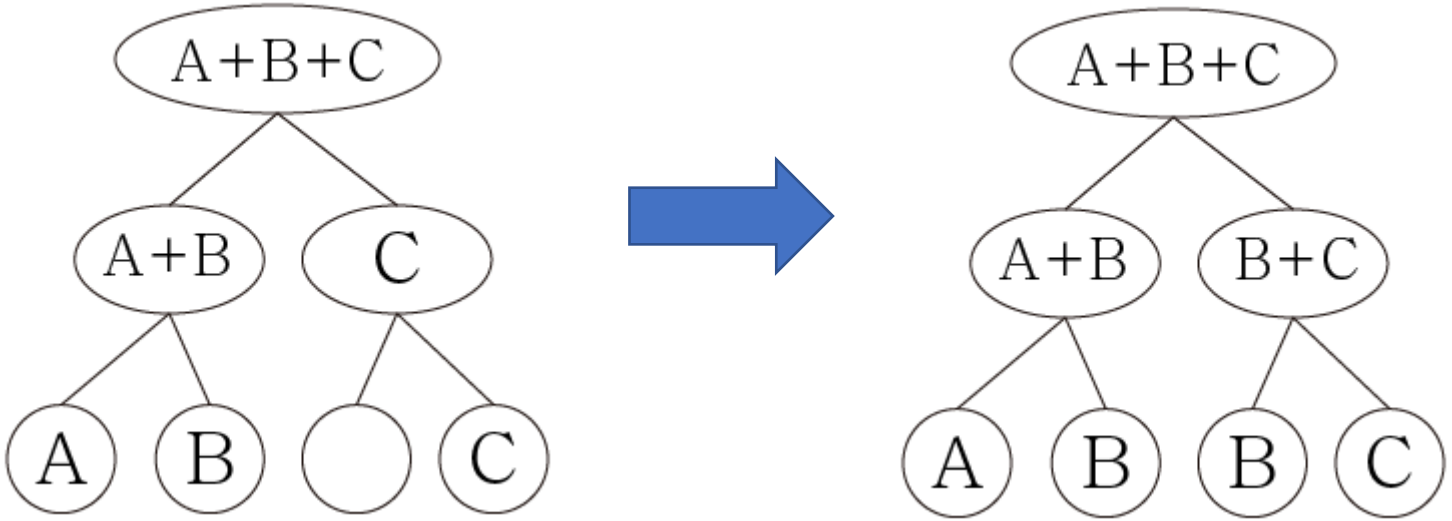
G 지역 꾸미기 게임

- 2d segment tree에 많은 정점을 빠르게 추가할 방법을 찾아보자.
- 추가된 정점은 옆 노드의 값을 복사해서 가져야 하기 때문에, 단순히 dynamic seg만으로는 부족하다.

- 1번 쿼리가 가로로 들어오는 경우에는 2d seg의 노드인 1d seg가 하나 통째로 복사되어야 하고, 세로로 들어오는 경우에는 2d seg의 노드인 모든 1d seg에 동일한 위치의 노드가 하나 복사되어야 한다.

G 지역 꾸미기 게임

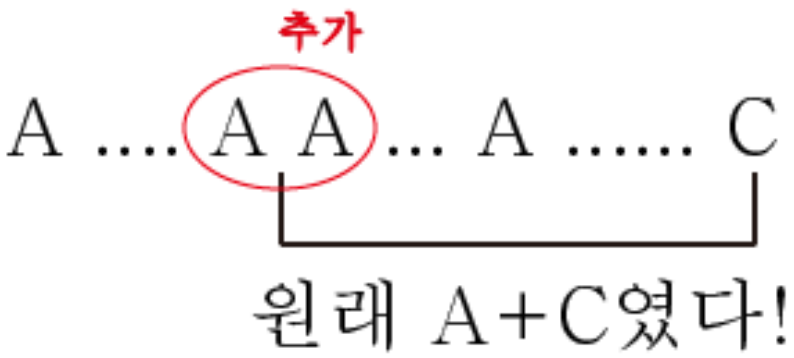
- 먼저 쿼리가 가로로 들어오는 경우부터 살펴보자.
- 세그먼트 트리 하나를 통째로 복제하는 연산은 원래부터 세그먼트 트리를 persistent하게 들고 있었다면 쉽게 할 수 있다.
- 그러나 아래 그림과 같은 경우, B+C에 해당하는 노드를 새로 만들어야 한다.



G 지역 꾸미기 게임

- 따라서 구간의 시작점에만 업데이트를 하는 것이 아니라 구간의 시작점과 끝점 모두에 업데이트를 해준다.

- 이러면 새로 추가되는 노드 딱 하나만 양산하면 나머지 추가되는 노드와 다른 노드가 잘 합쳐진 노드들은 전부 원래 있던 노드가 된다.



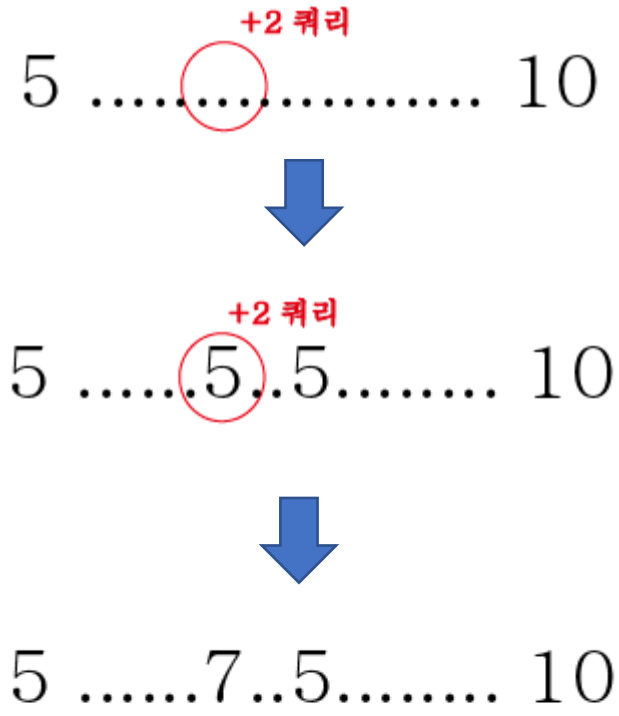
G 지역 꾸미기 게임

- 이제 쿼리가 세로로 들어오는 경우를 생각하자.
- 다행인 점은, 추가된 정점이 옆 노드와 같은 값을 가지기 때문에 3번 쿼리를 처리할 때는 추가된 정점이 있는지를 신경 쓸 필요가 없다는 것이다.
- 즉 2번 쿼리에서 어떤 정점을 방문할 때마다 방문하는 정점 하나만 추가해 줘도 된다.

G 지역 꾸미기 게임

- 세로 1번 쿼리를 받을 때 전역 set에 어떤 위치에 정점이 있어야 하는지를 저장해 두기만 하고 실제로 업데이트는 하지 않는다.
- 그 후 2번 쿼리를 받았을 때, 두 가지의 업데이트를 해야 한다.
- 먼저 자기 자신이 존재하지 않는 정점일 경우 자기 자신을 추가해야 한다.
- 이 때 존재하지 않는 정점의 값은 그 정점보다 왼쪽에 있는 존재하는 정점 중에 가장 오른쪽에 있는 정점의 값과 같다.
- 또 전역 set에서 봤을 때 자신의 바로 오른쪽에 있는 정점이 존재하지 않을 경우 그 정점을 추가해야 한다.
- 이는 “이 때 존재하지 않는 정점의 값은 그 정점보다 왼쪽에 있는 존재하는 정점 중에 가장 오른쪽에 있는 정점의 값과 같다.” 라는 조건을 여전히 만족해야 하기 때문이다.

G 지역 꾸미기 게임



H. Non-Decreasing Subarray Game

stonejjun03

H. Non-Decreasing Subarray Game (1)

- F(I,j)는 [i,j]의 non-decreasing subarray의 개수를 의미한다.
- 관찰1 : Platina는 당연히 L또는 R을 부른다.
- 관찰2 : 또한 이에 의해서 Yuto는 max(f(L,X),f(X,R))이 최소가 되는 X를 부를 것이다.
- 관찰3 : L,R에 대해서 특정 위치까지는 f(L,X)가 더 작을 것이며, 특정 위치부터는 f(X,R)가 더 작다.

H. Non-Decreasing Subarray Game (2)

- 관찰 4 : 관찰 2와 관찰 3에 의하여 Yuto가 부르게 되는 숫자는 더 작은 쪽이 변하게 되는 경계 둘 중 하나를 부르게 된다.

- 따라서 우리는 그 경계를 찾는 것이 목표이다.

H. Non-Decreasing Subarray Game (3)

- 특정 위치까지는 p 특정 위치 부터는 q이고 따라서 파라메트릭 서치를 이용하여 경계를 찾을 수 있다. 이 경계 한번을 찾는데 O(logN)에 해야하고, 따라서 F(i,j)를 O(1)에 찾아야 한다.

- K개 연속해서 non-decreasing하다면 그 구역에서 NDSA의 개수는 k(k+1)/2개다. 따라서 전처리로 이 구역별로 묶어줄 수 있다.

- Prefix sum을 이용하여 중간 몇 개 구역에서의 NDSA를 구할 수 있고, 완벽히 구역이 포함되지 않는 부분만 추가로 x(x+1)/2 로 더해주면 된다.

- 따라서 O(1)에 해결할 수 있고, 전체 과정을 O(QlgN)에 처리할 수 있다.

H. Non-Decreasing Subarray Game (4)

- 선형도 된다. O(N+Q) 도전해보시길!!

I. 조작된 ㄱ 폭탄 게임

stonejjun03

I. 조작된 ㄱ 폭탄 게임

1. PPT의 길이를 따분하게 2배로 늘리고 싶지 않습니다.
2. 진행과정에서 이미 블로그에 글을 작성하였습니다.
3. 블로그의 글을 PPT에 옮기기가 버거운 양입니다.
4. 3주간 계속 고민하여 나온 문제고, 블로그가 좀 활성화 되면 좋겠습니다.

그래서 죄송하지만 PPT에 작성하는 대신 블로그 링크 2개로 대체하겠습니다.

원론적/아이디어적 풀이 : https://stonejjun.tistory.com/71

사전지식을 이용한 풀이 : https://rkm0959.tistory.com/139