

20.08.26 동아리 풀이

선린인터넷고등학교 소프트웨어과

30610 나정휘

<https://JusticeHui.github.io>

문제 목록

- A. 반복하지 않는 수
- B. 극장 좌석 - 한국 올림피아드 2005 초등부 3번
- C. 준오는 심술쟁이야 - 머그컵 A번
- D. 두 섬간의 이동 - 제1회 KrijiCPC 2번
- E. 교수님은 기다리지 않는다 - ICPC 일본 리저널 2012 F번
- F. 나무 말고 꽃 - ICPC 독일 리저널 2013 E번
- G. 휴대폰 자판 - ICPC 남미 리저널 2012 C번
- H. XOR MST
- I. Ploughing - 폴란드 올림피아드 2005/2006 Stage 2 6번

반복하지 않는 수 (1)

- 반복하지 않는 수 100만 개 구한 다음 출력하면 된다.
- <http://boj.kr/2248e8c6cb6549ee9218f9f6805d0ee9>

극장 좌석 (1)

- vip석으로 인해 나뉜 조각들에 대해
- 각각 답을 구해서 곱하면 된다.
- 크기가 N인 조각의 답은 N번째 피보나치 수이다.
 - $F(1) = 1, F(2) = 2, F(3) = 3, \dots$
- <http://boj.kr/0c51f94f8f31404894f0281a752be585>

준오는 심술쟁이야 (1)

- $D(i, j) := i$ 번째 문자까지, k 들의 합이 j 인 경우의 수
- $D(i, j) = D(i-1, j-25) + D(i-1, j-24) + D(i-1, j-23) + \dots + D(i-1, j)$
 - DP배열의 각 행마다 누적합을 구해주면 $O(NS)$ 에 문제를 풀 수 있다.
- <http://boj.kr/5eebb8650ffd4c6a9b3e811ebf9d1de3>

두 섬간의 이동 (1)

- Union-Find를 하면서 각 집합의 크기를 저장하자.
- 간단한 식 계산을 통해 문제를 풀 수 있다.
- 크기가 a, b 인 두 집합이 연결된다면
 - 왕래가 가능한 쌍의 개수는 $f(a+b) - f(a) - f(b)$ 만큼 변한다.
 - 최소 다리 개수 합은 $S(a+b) - S(a) - S(b)$ 만큼 변한다.
 - 단, $f(x) = xC2$, $S(x) = f(1) + f(2) + \dots + f(x)$ 이다.
- <http://boj.kr/32983e5ef22943b69476dd4c376866b9>

교수님은 기다리지 않는다 (1)

- Unknown은 Union-Find를 이용해 쉽게 처리할 수 있다.
- 두 물건의 무게 차이는 트리에서 두 정점의 거리와 동일하다.
 - 루트 정점을 r 이라고 하자.
 - b 가 a 보다 얼마나 무거운지 계산하는 것은
 - $(r$ 이 a 보다 얼마나 무거운지) + $(b$ 가 r 보다 얼마나 무거운지) 계산한 결과가 되고
 - r 대신에 a 와 b 의 lca 를 사용해도 상관 없기 때문에 거리와 동일하다.
- <http://boj.kr/fa1da7ae94fe4ec69a94c6f88844fed5>

나무 말고 꽃

- 회전체의 부피는 적분으로 구할 수 있다.
- e^{-x^2} 를 적분할 수 있나...
- 구분구적법을 잘 구현해주면 된다.
- 구데기 문제 올려서 죄송합니다.
- <http://boj.kr/3f76b3ac0eef4f77ae18f8736ab1f547>

휴대폰 재판 (1)

- 트라이를 잘 구현하면 된다.
- <http://boj.kr/515de9f1cb6640938b8daa493e547fe5>

XOR MST (1)

- 각 가중치는 30bit로 표현할 수 있다.
- xor로 표현되는 가중치를 최소화하는 것이 목적이니, Binary Trie를 생각해볼 수 있다.

XOR MST (2)

- Boruvka Algorithm 비슷하게 MST를 구축한다.
- 30번째 비트가 켜져 있는 정점 집합과 꺼져 있는 집합을 잇는 간선은 한 개만 있는 것이 최적이다.
- 켜져 있는 원소들로 Binary Trie를 만든 뒤, 꺼져 있는 원소와 XOR했을 때 최소가 되는 값을 찾아서 연결하자.

XOR MST (3)

- 30번째 비트가 다른 원소 집합은 서로 연결이 되었다.
 - 이제 두 집합에 대해 각각 분할한 다음 문제를 풀어도 된다.
- 두 집합으로 분할한 다음 29번째 비트에 대해 문제를 풀고
- 분할하고
- 28번째 비트에 대해 풀고
- 분할하고
- 반복하면 문제를 풀 수 있다.
- <http://boj.kr/e42aa1312ba342fa85bfdb21713f3161>

Ploughing (1)

- 열심히 직사각형을 깎아내다 보면
- 마지막에는 가로 혹은 세로 길이가 1인 직사각형을 없애게 된다.
 - 그 직사각형의 크기를 $1 * K$ 라고 하면
 - 직사각형을 총 $N+M-K-1$ 번 깎아내게 된다.
 - $N+M-1$ 는 일정하므로 K 를 최대화하는 것이 목적이다.

Ploughing (2)

- 마지막에 가로로 길쭉한 직사각형을 남긴다고 하자.
 - 세로로 길쭉한 직사각형을 남기는 경우는 배열을 회전시켜서 수행
- 마지막까지 남겨놓을 직사각형의 양 끝 점을 고정하면
 - 그런 직사각형을 남겨놓을 수 있는지 $O(N+M)$ 에 판별할 수 있다.
 - 총 $O(M^2)$ 가지의 경우가 있기 때문에 복잡도는 세제곱이다.
- 최적화하자!

Ploughing (3)

- 투표인터 느낌으로 끝 지점을 관리하자.
- 총 $O(M)$ 개의 경우만 고려해주면 되고, 제공 시간에 문제를 풀 수 있다.
- <http://boj.kr/f1368fdd6e164147995af83fa871f74b>