20.11.04 동아리 풀이

선린인터넷고등학교 소프트웨어과
30610 나정휘
https://JusticeHui.github.io

문제 목록

- A. 소수인 팰린드롬
- B. 알약 - 2011 로키산맥 리저널 E번
- C. 개똥벌레 - 2007 크로아티아 올림피아드 예선 3번
- D. 안티 팰린드롬
- E. 날카로운 눈 - 2007 SEERC E번
- F. 트리의 색깔과 쿼리 - 2019 UniCon 9번
- G. 타일 놓기 - SRM 383 Div.1 Med / CF 668 Div.1 E
- H. 데이터 제작 - 2020 UCPC K번

ㅎㅎㅎ

- 제가 만든 문제를 2개 넣었습니다.
- 출제자의 풀이를 직접 볼 수 있는 아주 좋은 기회이니
- **꼭 푸세요ㅎㅎ**

- F. 트리의 색깔과 쿼리 - 2019 UniCon 9번
- H. 데이터 제작 - 2020 UCPC K번

소수인 팰린드롬 (1)

- 두 가지 방법을 생각해볼 수 있다.
  - 모든 소수를 구한 뒤 팰린드롬 판별
  - 모든 팰린드롬을 구한 뒤 소수 판별

- 1부터 N까지
  - 소수 개수는 O(N / log N) - 소수 정리
  - 팰린드롬 개수는 O(√N)

- 모든 팰린드롬을 구한 뒤 소수 판별이 이득

소수인 팰린드롬 (2)

- 모든 팰린드롬을 구하는 것
  - 1글자 혹은 2글자로 된 문자열에서 시작
  - 양 끝에 동일한 문자를 붙인다.
  - **BFS**

- BFS로 모든 팰린드롬을 구한 뒤 소수 판별을 해주면 된다.

- http://boj.kr/6afe82ab30f2417c9617e542c67865ac

알약 (1)

- D(i, j) = 온전한 알약이 i개, 반쪽짜리 알약이 j개 있을 때 정답
- D(i, j) = D(i-1, j+1) + D(i, j-1)
  - 음수 인덱스 예외처리
- http://boj.kr/6f53e7c0bf74484880a7c83ba8281ece

개똥벌레 (1)

- 구간 [Si, Ei]에 1을 더한 뒤, 최댓값을 찾는 문제

- 세그먼트 트리를 쓴다?
  - 이래서 뉴비한테 세그같은 거 알려주면 안 됩니다 ——

- 누적합 배열(변홧값 배열)을 알아보자.

개똥벌레 (2)

- 구간에 어떤 수를 더하는 쿼리가 **모두** 들어온 다음
- 어떤 위치에 대한 값을 물어보는 형태의 문제에 사용

- 구간 [Si, Ei]에 Vi를 더한다고 하면
  - `A[Si] += Vi; A[Ei + 1] -= Vi;`
- 하고 누적합을 쭉 구해주면 된다.

개똥벌레 (3)

- A[Si] += Vi 로 인해서 Si부터 끝까지 Vi가 더해진다.
- A[Ei + 1] -= Vi로 인해서 Ei+1부터 끝까지 Vi가 빠진다.
- 결과적으로 Si부터 Ei까지만 Vi가 더해진다.

- http://boj.kr/fa2b7902175e4c66924f00aa771e8dd7

안티 팰린드롬 (1)

- 가장 많은 알파벳이 (N+1)/2개 이상이면 불가능하다.
  - 비둘기집의 원리에 의해, 최소한 한 위치는 겹친다.

- 그렇지 않다면 항상 가능하다.
  - 앞쪽 (N+1)/2개는 사전순으로 찍고
  - 그 다음에는 겹치지 않게 최대한 사전순으로 앞서도록 찍으면 된다.

- http://boj.kr/8386b7d8c3304f7ca146e78aee5cf65e

날카로운 눈 (1)

- **"수의 개수"**에 대한 누적합을 생각해보자.

- 홀수 번 나오는 원소가 한 개 존재한다면
  - 누적합이 짝수에서 홀수로 바뀌는 지점이 한 곳 존재한다.
- 홀수 번 나오는 원소가 존재하지 않는다면
  - 누적합은 항상 짝수다.
- 누적합이 처음으로 홀수가 되는 지점을 Parametric Search로 찾으면 된다.
- http://boj.kr/a10eab6d0e304a0391b560300366d220

트리의 색깔과 쿼리 (1)

- 간선을 제거하는 쿼리는 순서를 뒤집어서 처리하면 간선을 추가하는 쿼리로 생각할 수 있다.
  - Union Find
- 서로 다른 색깔의 개수는 std::set으로 관리할 수 있다.

트리의 색깔과 쿼리 (2)

- Union Find를 하면서 Union할 때 두 컴포넌트의 set을 합쳐주면 된다.
- 대충 합치면 원소의 이동 횟수가 O(N^2)이다.

- 작은 set에 있는 원소를 큰 set으로 이동하는 방식으로 합쳐주면 최대 O(N log N)번 이동한다.

트리의 색깔과 쿼리 (3)

- 이동 횟수가 O(N log N)임을 증명하자.
  - 편의상 multiset이라고 생각한다.
  - set에서는 중복 원소가 없어지기 때문에 multiset보다 이동횟수가 많아지지 않는다.

  - 각 원소는 처음에 크기가 1인 집합에 있다.
  - 크기가 각각 A, B인 원소를 합치면 A+B가 된다.
  - 작은 집합에 있는 원소를 큰 원소로 옮기면 **"옮겨지는 원소"**가 속하는 집합의 크기가 **2배 이상**이 된다.
  - 집합의 크기는 N을 넘을 수 없기 때문에 각 원소는 최대 O(log N)번 이동한다.

트리의 색깔과 쿼리 (4)

- 이 기법을 Smaller to Larger라고 부른다.
- O(N log^2 N + Q)정도에 문제를 풀 수 있다.

- http://boj.kr/7743c7b37ccc4b39bfb520b0cc4d36ec

타일 놓기 (1)

- 타일의 개수를 최소화한다는 것은
  - 모서리를 최대한 많이 끊어내는 것으로 생각할 수 있다.
  - 이때 정답은 (타일을 놓아야 하는 칸의 개수) - (끊은 모서리 개수)

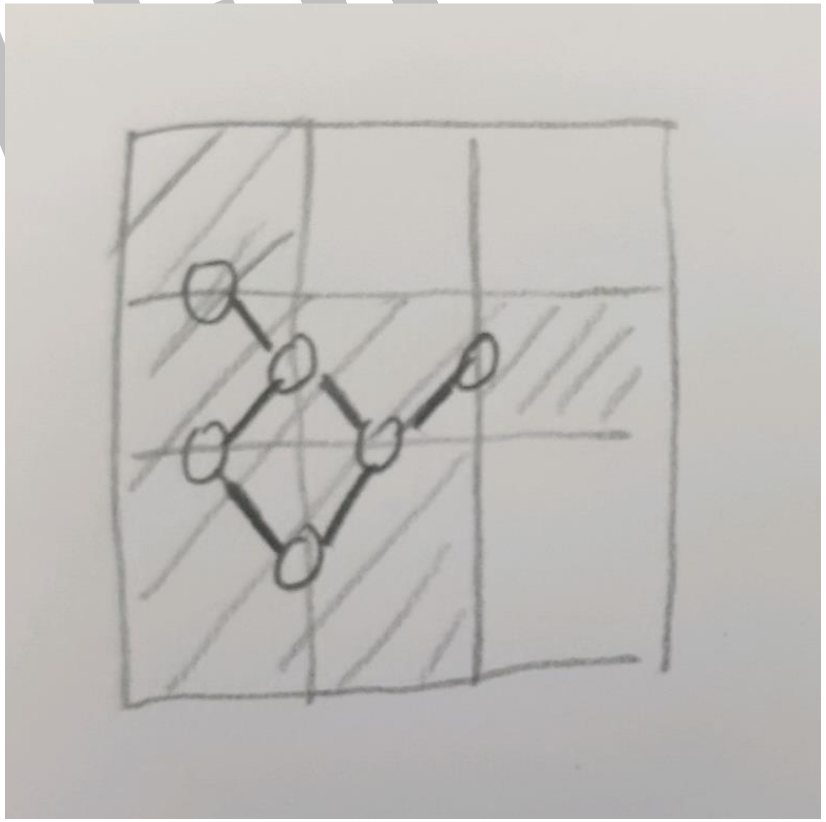
- 타일은 1*K 꼴이기 때문에 각 덩어리가 꺾이면 안 된다.
  - L 모양이 되도록 모서리를 끊으면 안 된다.

타일 놓기 (2)

- 격자의 각 모서리를 정점으로 하고, 끊으면 L 모양이 나오는 두 정점을 간선으로 연결하자.

- 선택한 정점이 인접하지 않도록
- 최대한 많은 정점을 선택하면 된다.

- **최대 독립 집합**



타일 놓기 (3)

- 만들어진 그래프는 이분 그래프이다.
  - 최대 독립 집합의 크기는 (정점 개수) - (최대 매칭 크기)
    - Kőnig's theorem
  - Dinic이나 Hopcroft-Karp를 짜면 $O(NM\sqrt{(NM)})$에 문제를 풀 수 있다.

- http://boj.kr/2a5e0503718549058cd338d378868a58

데이터 제작 (1)

- 국가대표가 인정한 좋은 문제!
- UCPC 2020 실시간 방송 풀이 영상 : https://youtu.be/ilhvcHJ8aQY?t=3180

- 문제를 간단하게 요약해보자.
  - 정점 N개, 간선 M개, 면 K개인 평면 그래프 제작 (outer face 제외)
  - self loop와 multi edge 허용 X
  - **(중요) 좌표 범위 제한 있음**

- 좌표 범위 제한만 없으면 아주 쉬운 문제인데
- 좌표 제한 때문에 어려운 문제가 되었다.

데이터 제작 (2)

- V, E, F, C를 각각 평면 그래프의 정점, 간선, 면, 컴포넌트의 개수라고 하면
- V-E+F = C+1이 성립한다.
  - N = V, M = E, K = F-1이므로 N-M+K = C이다.

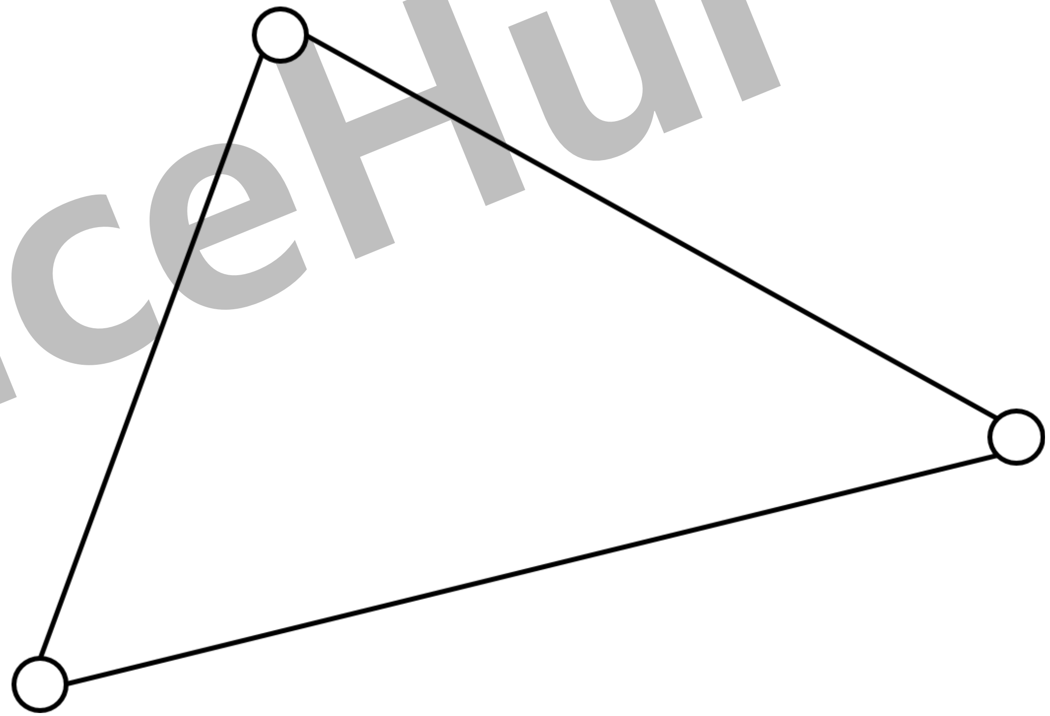
- self loop와 multi edge가 없는 평면 그래프에서는 M ≤ 3N-6이 성립한다.
- 그러므로 N개의 정점과 3N-6개의 간선을 좁은 공간에 밀어넣는 방법을 찾은 뒤
- M < 3N-6이라면 간선을 적당히 제거해주면 된다.

- C ≠ 1인 경우에는 정점 C-1개를 따로 빼내면 C = 1인 상황으로 만들 수 있다.

데이터 제작 (3)

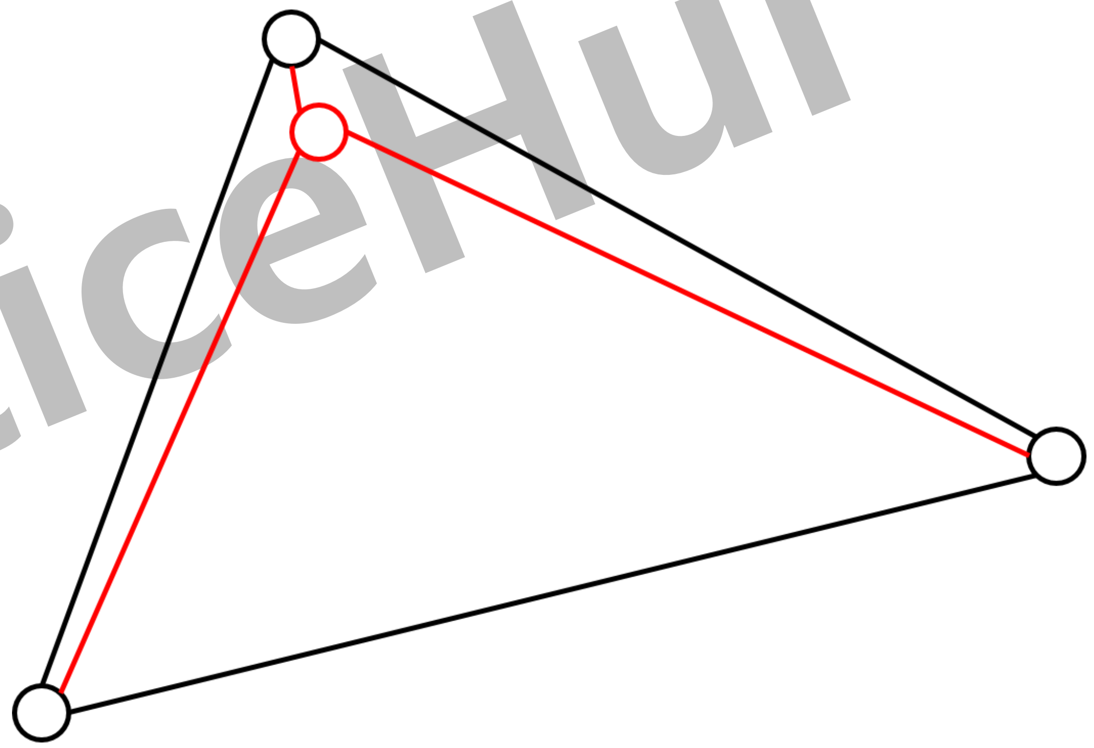
- 좌표 범위를 신경 쓰지 않고, 일단 3N-6개의 간선을 만드는 방법을 알아보자.

- 먼저, 충분히 큰 삼각형에서 시작한다.
- 정점은 3개, 간선은 3(= 3*3-6)개 있다.



데이터 제작 (4)

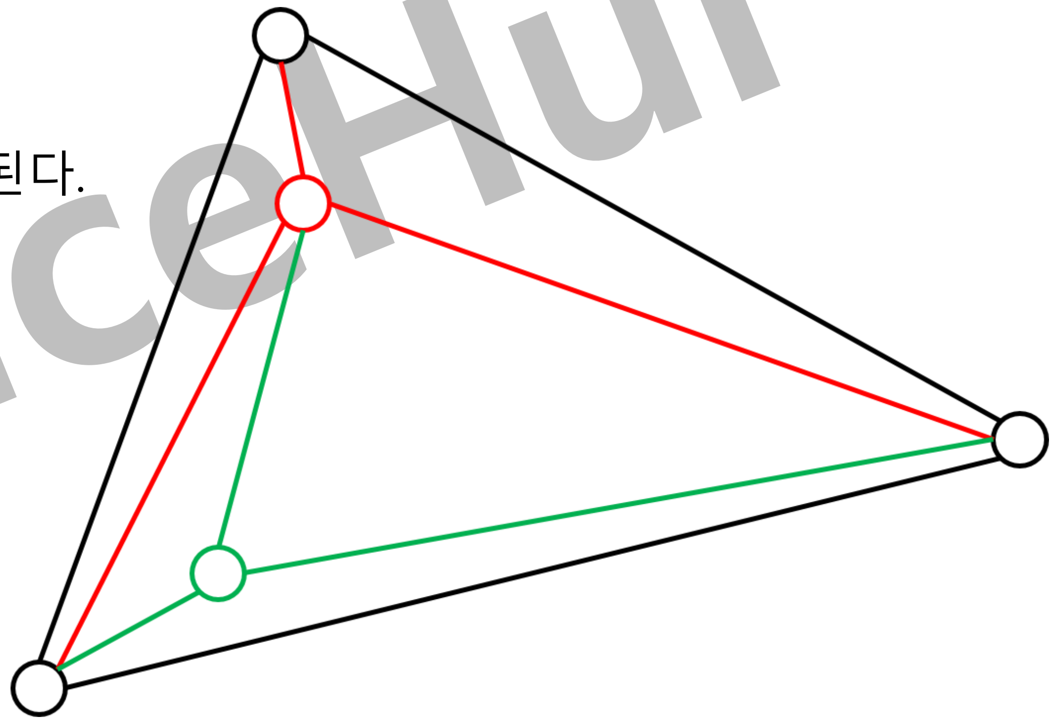
- 전 단계에서 만든 삼각형 내부에 점 하나를 찍으면
- 정점 1개와 간선 3개를 만들 수 있다.



데이터 제작 (5)

- 점 하나를 추가해서 만들어진 삼각형 내부에 점을 추가하면
- 정점 1개와 간선 3개를 또 만들 수 있다.

- 점이 N개 생길 때까지 이 과정을 반복하면 된다.



데이터 제작 (6)

- 모든 영역을 삼각형으로 만들면 간선이 3N-6개가 된다는 것을 알아냈다.

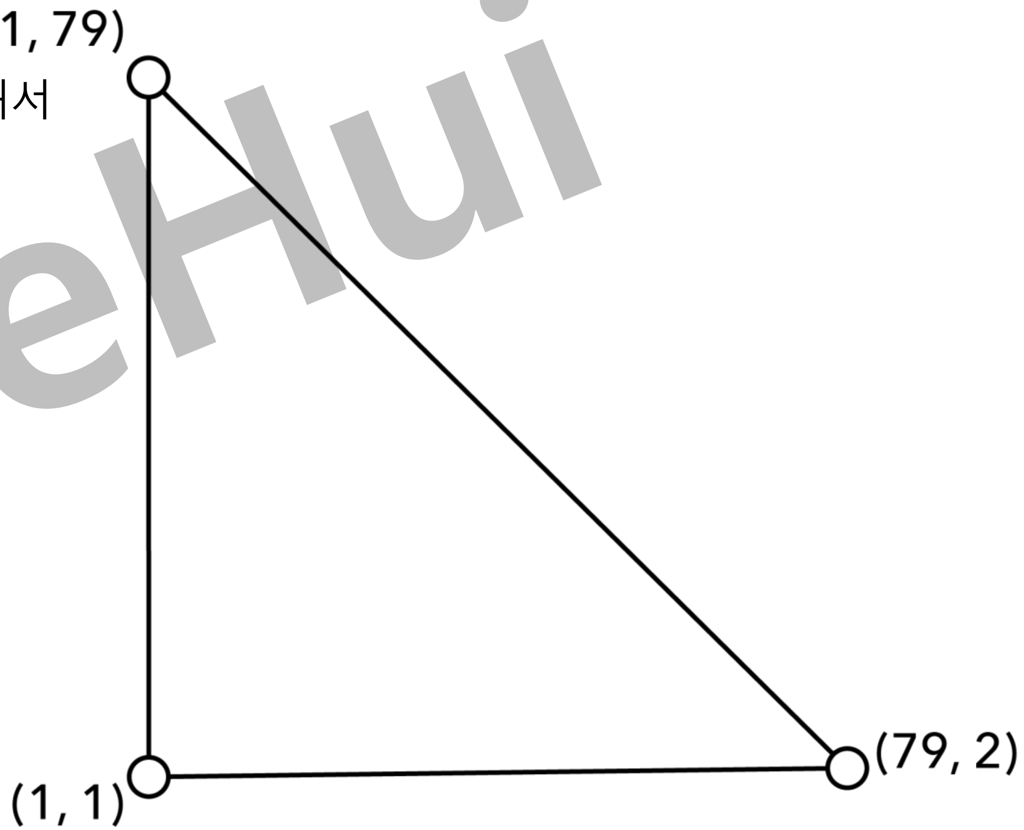
- 이제, 최대한 좁은 공간에 넣어보자.
- 정점 3개로 큰 삼각형을 만들고
- 그 안에 나머지 N-3개의 점을 넣을 수 있어야 한다.

데이터 제작 (7)

- 큰 삼각형을 만들자.
  - (79, 1)이 아닌 이유는 간선의 기울기를 약간 다르게 해서
  - 간선이 겹치는 것을 피하기 위함이다.

- 삼각형 내부에
- y = 78인 점 1개
- y = 77인 점 2개
- …
- y = i인 점 (79 - i)개

- 총 3 + (78 * 79) / 2 = 3084개의 점을 만들 수 있고
- 이는 3000개의 점을 넣기에 충분하다.



데이터 제작 (8)

- 3N-6개의 간선을 이용해 삼각형으로 쪼개는 것은 오른쪽 그림처럼 하면 된다.

- 빨간색 간선은 Connected Graph를 만들기 위해 꼭 필요한 간선이고
- 파란색 간선은 M의 값에 따라 제거해도 되는 간선이다.

- ~~그냥 점을 잘 뿌리고 들로네 삼각분할을 해도 되더라…~~

- http://boj.kr/42ecf90ef0c5458e823b76608f3e492e