

20.11.11 동아리 풀이

선린인터넷고등학교 소프트웨어과

30610 나정휘

<https://JusticeHui.github.io>

문제 목록

- A. Fence Planning - USACO 2019 US Open Silver 3번
- B. 조 짜기
- C. 비밀번호
- D. Visible Lattice - 2009 인도 리저널 F번
- E. 숫자의 신
- F. 2172 팰린드롬 경로
- G. 12106 부분 문자열의 개수
- H. Mountains - IOI 2017 연습세션 1번

Fence Planning (1)

- 단순 구현 문제
- DFS / BFS / Union Find
- 다양한 선택지가 있다.
- <http://boj.kr/5bd018b6bba64584aa8403dd3b90194f>

조 짜기 (1)

- $D(i)$ = 1~i번째 학생으로 조를 짤 때 최댓값
- $D(i) = \text{Max}\{ D(j) + \max(j+1, i) - \min(j+1, i) \}$
- 모든 구간 $[i, j]$ 의 최대/최솟값은 $O(N^2)$ 만에 전처리할 수 있다.
- $O(N^2)$ 에 문제를 풀 수 있다.
- <http://boj.kr/02c5dda7343b4f20b850b4ddcd80c75f>

비밀번호 (1)

- $D(i, j)$ = 길이가 i 이고 마지막 수가 j 인 비밀번호의 경우의 수
- $D(i, j) = \sum D(i-1, k)$
 - k 는 j 와 인접한 수
- $O(N)$ 에 문제를 풀 수 있다.
- <http://boj.kr/7b325956d18141f28618e4c943570f9e>

Visible Lattice (1)

- KOI 2014 고등부 1번 관중석과 비슷한 느낌
- i, j, k 가 모두 0 이상 N 이하인 정수일 때
- $\gcd(i, j, k) = 1$ 이 되는 경우의 수를 세는 문제
- 열심히 구현하면 된다.
- <http://boj.kr/3591b306999143ecb703700ecab3d15e>

숫자의 신 (1)

- 일단 각 수를 한 번씩은 사용하고
- K-N개를 추가로 더 사용해야 한다.
- 추가로 더 사용하는 수는
 - 길이가 길수록 좋고
 - 길이가 같다면 사전순으로 더 뒤에 있는 수가 좋다.
 - 그런 수를 찾아서 K-N개 추가하면 된다.
- 이제, K개의 수를 잘 배치해서 결과를 최대화 시키면 된다.

숫자의 신 (2)

- **Theorem.** 수 A, B 가 있을 때 $\text{concat}(A, B) > \text{concat}(B, A)$ 가 되도록 정렬하는 것이 최적이다.
- **Proof.** Exchange Argument로 증명할 수 있다. 직접 해보자.
- <http://boj.kr/3eba6c94d29b4a9daf8831246ed58265>

팰린드롬 경로 (1)

- $D(i1, j1, i2, j2, len)$
 - $(i1, j1)$ 에서 시작하고 $(i2, j2)$ 에서 끝나는
 - 길이가 len 인 팰린드롬의 개수
- 길이가 1 또는 2인 경우를 base case로 잡고
- 각 상태 당 64가지의 상태 전이를 보면 된다.
- $O(N^4 L)$ 에 문제를 풀 수 있다.
- <http://boj.kr/0186cef2ff1c4ce495bbf3183ab3d5f0>

부분 문자열의 개수 (1)

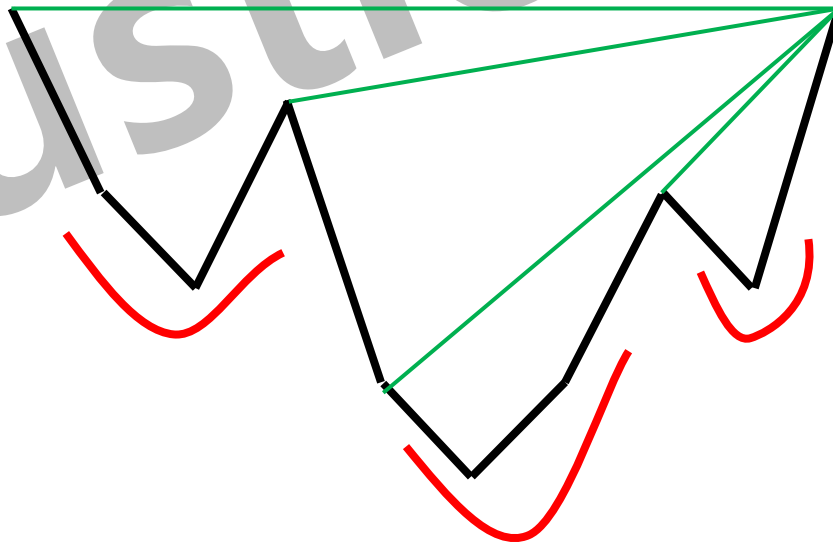
- $D(i, j, c)$
 - 만든 문자열의 i 번째 글자가 c 이면서
 - 패턴(주어진 문자열)의 j 번째 글자까지 매칭되는 경우의 수
- 만든 문자열의 i 번째 글자와 패턴의 j 번째 글자가 매칭됐는데
- $i+1$ 과 $j+1$ 이 매칭이 안 된다면?
 - KMP의 Failure Function을 따라가면 된다.
- KMP + DP로 풀 수 있다.
- <http://boj.kr/b1c69aab4ff047f79feeffdd5f306735>

Mountains (1)

- 오른쪽에 있는 i 가 왼쪽에 있는 j 를 볼 수 있는 조건은?
 - j 초과 i 미만인 모든 k 에 대해 $ccw(j, k, i) \geq 0$
 - 이해가 안 된다면 그림을 그려보자.
- $D(j, i) = [j, i]$ 구간의 답이라고 정의하자.
- Naïve하게 계산하면 $O(N^3)$ 이고 당연히 TLE.

Mountains (2)

- i 번째에서 볼 수 있는 곳을 생각해보자. (초록색 직선)
- 볼 수 있는 곳을 기준으로 분할된 구간에 대해서
- 각각 독립적으로 문제를 풀어도 된다. (빨간색 곡선)



Mountains (3)

- i 를 고정하고, j 를 $i-1$ 부터 왼쪽으로 이동하면서 계산한다.
- 구간 $(j, i]$ 에서 i 가 볼 수 있는 가장 왼쪽 사람 $left$ 를 관리하자.
 - $ccw(j, left, i) \geq 0$ 이라면 i 는 j 를 볼 수 있으며
 - $(j, left)$ 사이에 있는 사람들은 $left$ 에 가려져서 안 보인다.
- 이 정보를 이용해 $ccw(j, left, i) \geq 0$ 을 만족하는 j 가 나올 때마다
- $D(j+1, left-1)$ 을 더해가면서 문제를 풀 수 있다.

Mountains (4)

- 각 i 에 대해 $left$ 는 최대 i 번만 움직인다.
- $O(N^2)$ 에 문제를 풀 수 있다.
- <http://boj.kr/f69b200d53d447b0817ad0b99f0d090a>