20.12.23 동아리 풀이

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

0 문제 목록

- 인형들 - 2018 카카오 코드 페스티벌 예선 B번
- 호반우 상인의 이상한 품질 계산법 - 2020 경북대학교 교내대회 F번
- 화살을 쏘자! - 2020 가톨릭대학교 교내대회 F번
- 호반우와 리듬게임 - 2020 경북대학교 교내대회 I번
- 등산 마니아 - 2020 한국정보올림피아드 2차 대회 중등부 2번
- L-트로미노 계단 - 2020 경북대학교 교내대회 L번
- 우체국 3
- 3D Histogram - 2020/2021 COCI Contest1 3번

1 인형들

단순 구현 문제다.

http://boj.kr/d561ee63735047c3bf0c68f849bfa2a0

2 호반우 상인의 이상한 품질 계산법

(1번째로 작은 것, 1번째로 큰 것), (2번째로 작은 것, 2번째로 큰 것), ... , (i번째로 작은 것, i번째로 큰 것)을 묶는 것이 최적이다.

왜 그런지는 생각해보자.

http://boj.kr/82183cb5c12847cf8cbe716a1a557909

3 화살을 쏘자!

사분면 / 기울기 별로 풍선의 개수를 구하면 된다.

실수 자료형은 값을 **정확하게** 표현하지 못하기 때문에 오차가 발생할 수 있다. 그러므로 기울기는 실수형보다 분수형으로 관리하는 것이 좋다.

http://boj.kr/7fea4db6284f474197af6f1a29dff662

4 호반우의 리듬게임

$D(i, j, k) = i$번째 노트를 칠 때 누적 콤보가 j이고, 연속으로 미스를 k번했을 때 얻을 수 있는 최대 점수
로 점화식을 정의해서 DP를 하자.

$N \geq 3$이라면 미스를 3번 연속으로 낼 수 있기 때문에 최댓값은 항상 0 이상이다.

http://boj.kr/28ac632b39814df2a63bf00b6f13d19b

5 등산 마니아

각 간선이 몇 번 사용되는지 알면 정답을 구할 수 있다.

정점 v와 v의 부모 정점을 연결하는 간선이 사용되는 횟수는, 트리에서 정점 2개를 고를 때 v를 루트로 하는 서브 트리 안에서 1개 이상 고르는 경우의 수와 동일하다.

v를 루트로 하는 서브 트리에 속하는 정점의 개수를 S_v라고 하자. 이때 v와 v의 부모를 연결하는 간선이 사용되는 횟수는 $S_v(S_v - 1) + S_v(N - S_v)$이다.

S_v는 $O(N)$에 전처리할 수 있으므로 전체 문제를 $O(N)$에 해결할 수 있다.

http://boj.kr/b74cd7a7206b4d47a463d756ae3f9545

6 L-트로미노 계단

N층짜리 계단을 만들기 위해서는 $N(N+1)/2$칸이 필요하고, 이 수는 3의 배수가 되어야 한다.
그러므로 N을 3으로 나눈 나머지가 1이라면 불가능하다.

N이 작은 경우($N < 10$) 가능한지 불가능한지 여부는 다음과 같다.

- $N = 2, 6, 8, 9$: 가능
- $N = 3, 5$: 불가능

$N \geq 10$인 경우에는 N을 3으로 나눈 나머지가 1이 아니라면 아래와 같이 재귀적으로 타일을 배치하면 된다.

- N을 3으로 나눈 나머지가 0인 경우
  - 위에 크기 6인 계단
  - 오른쪽에 크기 $N - 6$인 계단
  - 가운데 높이 $N - 6$, 너비 6인 직사각형
- N을 3으로 나눈 나머지가 2인 경우
  - 위에 크기 $N - 2$인 계단
  - 오른쪽에 크기 2인 계단
  - 가운데 높이 2, 너비 $N - 2$인 직사각형

N이 2, 6, 9인 경우만 손으로 정답을 만들어주면 나머지는 재귀적으로 처리할 수 있다.

http://boj.kr/7c32ab67211e41d088dcbad84c6ec3e2

7 우체국 3

편의상 디스크립션에 나와있는 V, P, L 대신 N, K, L이라는 문제로 표기한다.

원형으로 구성된 배열을 적절히 자르는 문제다. 원형 구조를 다루는 것은 매우 귀찮기 때문에 한 곳을 끊어서 선형으로 만들자.
선형에서는 간단한 2차원 DP로 문제를 해결할 수 있다:

- $D(K, N) = 1 \ldots N$번째 원소를 K개의 조각으로 잘랐을 때의 최소 비용
- $D(K, N) = \min(D(K-1, i) + C(i+1, N))$

구간 $[i, j]$의 비용 $C(i, j)$는 구간의 중앙값으로 가는 거리의 합이 되고, 이는 Prefix Sum을 미리 전처리해두면 $O(1)$에 계산할 수 있다.
그러므로 DP를 Naive하게 계산해주면 $O(KN^2)$이다. 끊을 수 있는 지점이 N개 있으므로 모든 경우를 다 보면 $O(KN^3)$이다.

$C(i, j)$는 사각 부등식을 만족하기 때문에 **Divide and Conquer Optimization**을 사용할 수 있다.
$O(KN^2)$ DP를 $O(KN \log N)$으로 최적화할 수 있으므로 전체 시간 복잡도는 $O(KN^2 \log N)$이 되어 문제를 해결할 수 있다.

http://boj.kr/f234a838d12c4854a7e90906df7f3524

8 3D Histogram

2차원의 경우는 매우 유명한 문제이고, Monotone Stack을 이용한 풀이와 분할 정복을 이용한 풀이가 잘 알려져 있다. Monotone Stack을 3차원에 적용하는 것은 어려울 것 같으니 분할 정복으로 접근해보자.

구간 $[s, e]$에 속한 히스토그램을 처리할 때, 중점 $m = \lfloor \frac{s+e}{2} \rfloor$을 지나는 모든 히스토그램을 처리한 뒤, 구간 $[s, m-1]$과 $[m+1, e]$에 대해 재귀적으로 처리할 것이다.

함수 $f(i, j)$와 $g(i, j)$를 아래와 같이 정의하자.
$f(i, j) = \min_{i \le k \le j}(a_k)$
$g(i, j) = \min_{i \le k \le j}(b_k)$

구간의 양 끝점 s, e와 중간 지점 m에 대해, 구간의 상태를 아래 4가지로 분류할 수 있다.

1. $f(s, m) \le f(m, e) \cap g(s, m) \le g(m, e)$
2. $f(s, m) \le f(m, e) \cap g(s, m) \ge g(m, e)$
3. $f(s, m) \ge f(m, e) \cap g(s, m) \le g(m, e)$
4. $f(s, m) \ge f(m, e) \cap g(s, m) \ge g(m, e)$

1, 2를 처리할 수 있으면, 적절히 reverse하고 swap하는 것으로 3, 4번을 똑같이 처리할 수 있다. 1, 2번 케이스에 대한 풀이를 알아보자.

1번 케이스는 투포인터를 이용해 $O(e - s + 1)$에 해결할 수 있다.
각 시작점 l에 대해, $f(l, r) = f(l, m) \cap g(l, r) = g(l, m)$인 r의 최댓값을 찾아서 최대 부피를 갱신해주면 된다. (정답 코드에서 calc1 함수 참고)

2번 케이스는 다소 복잡하다. $(r - l + 1)f(l, m)g(m, r)$을 최대화하는 것이 목적이다.
식을 조금 변형하면, $f(l, m)g(m, r)(1 - l) + f(l, m)g(m, r)r$이 된다.
$f(l, m)$으로 묶어주면, $f(l, m) \times (g(m, r)(1 - l) + g(m, r)r)$이 된다.

중괄호 안쪽의 식은 $g(m, r)$이 기울기, $g(m, r)r$이 y절편, $(1 - l)$이 변수인 일차 함수로 해석할 수 있다. 일차 함수가 여러 개 있을 때 임의의 x좌표에서 최댓값을 구하는 것은 **Convex Hull Trick**을 이용해서 빠르게 구할 수 있으니, 이 성질을 이용해서 풀이를 찾아보자.

먼저 1번 케이스처럼 각 시작점 l에 대해, $f(l, r) = f(l, m) \cap g(l, r) = g(l, m)$이 되도록 하는 r의 **구간**을 전처리하자. $O(e - s + 1)$ 시간이 걸린다. 이 구간의 왼쪽 끝점과 오른쪽 끝점을 각각 $L(l), R(l)$이라고 하자.

각 시작점 l에 대해, $L(l)$부터 $R(l)$번째 직선들 중에서 $(1 - l)$ 시점에서의 최댓값을 구하면 된다. 이것은 Segment Tree의 각 정점에서 Convex Hull Trick을 관리하는 것으로 처리할 수 있다.
Segment Tree의 각 정점에 저장되는 직선의 기울기, 쿼리로 주어지는 x좌표인 $(1 - l)$ 모두 단조성이 있기 때문에 Convex Hull Trick을 선형에 처리할 수 있다.

전체 시간 복잡도는 $T(N) = 2T(N/2) + O(N \log N)$이 되므로 $O(N \log^2 N)$에 문제를 풀 수 있다.

http://boj.kr/be6364b253e44ef0869295481796ed19