

방학 연습문제 풀이 - Week 1

정덕인

[정답 코드 보기](#)

BOJ 1087 - 쥐 잡기

- solved.ac 기준 난이도: Platinum IV
- 분류: 삼분 탐색

BOJ 1087 - 쥐 잡기

- 모든 쥐가 한 번에 잡히지 않는 최대 L 을 구하는 문제
- 쥐가 고정되어 있으면, 그 때 모든 쥐가 한 번에 잡히지 않는 최대 L 은 어떻게 구할까?
- $L = \max((\text{쥐의 } x\text{좌표 최대}) - (\text{쥐의 } x\text{좌표 최소}), (\text{쥐의 } y\text{좌표 최대}) - (\text{쥐의 } y\text{좌표 최소}))$

BOJ 1087 - 쥐 잡기

- 문제는 쥐가 고정되지 않고 **등속 운동**한다.
- 어떠한 시간 t 에서 L 의 값을 함수 $L(t)$ 로 나타내자.
- 이때 $L(t)$ 를 구할 수 있을까?
- **구할 수 있다!**

BOJ 1087 - 쥐 잡기

- 쥐가 고정되어 있을 때 L의 값을 생각하자.
- (쥐의 x좌표 최대) - (쥐의 x좌표 최소) =
$$\max((p_{xi} + v_{xi} * t) - (p_{xj} + v_{xj} * t))$$
- y좌표도 똑같이 하면 된다.

BOJ 1087 - 쥐 잡기

- 그런데 \max 안의 식을 정리하면, t 에 대한 일차식임을 알 수 있다.
- $L(t)$ 는 여러 일차함수들의 최댓값이므로, **아래로 볼록한 함수임을** 알 수 있다!
- 이제 $L(t)$ 의 최솟값을 구하면 해결.

BOJ 1087 - 쥐 잡기

- 최솟값은 어떻게 구할까?
- **삼분탐색**을 이용!
- 자세한 내용은 [관련 블로그](#) 참고
- 삼분탐색을 이용하면 $O(N^2)$ 시간복잡도로 해결할 수 있다.

BOJ 13907 - 세금

- solved.ac 기준 난이도: Platinum V
- 분류: 다익스트라

- 도로마다 통행료가 고정되어 있다면?
- 단순 다익스트라 한 번으로 해결 가능!
- 그런데 통행료가 업데이트 될 때마다 다익스트라를 돌리면?
시간복잡도 $O(KM \log N)$ 으로 TLE.

- 통행료 업데이트가 모두에게 동일하게 적용됨을 이용하자.
- $D(i, j) := i$ 번 도시까지 j 개의 도로를 거쳐 갈 경우의 최소 비용
- 이때 각 도로마다 통행료가 t 만큼 오르면, $D(i, j)$ 는 jt 만큼 오르게 된다.

- $D(i, j)$ 는 어떻게 계산할까?
- 다익스트라 알고리즘을 이용하면 된다.
- 어떤 정점까지의 최단경로가 가질 수 있는 간선의 최대 개수는 $N - 1$ 개이므로, N^2 개의 정점에서 다익스트라를 돌린다고 생각하면 됨.
- 시간복잡도 $O(N^2 + M \log N)$ 으로 $D(i, j)$ 계산 가능

- 이제 K 개의 질의에 대해, $O(N)$ 으로 처리할 수 있으므로
- 시간복잡도 $O(N^2 + M \log N + KN)$ 으로 전체 문제를 해결할 수 있다.

BOJ 1280 - 나무 심기

- solved.ac 기준 난이도: Platinum IV
- 분류: Fenwick Tree

BOJ 1280 - 나무 심기

- i 번 나무를 심는 데 드는 비용을 $cost(i)$ 라 하자.
- $cost(i) = \sum_{j < i} |X_i - X_j|$
- 단순히 계산하면 하나의 $cost(i)$ 를 계산하는 데에 $O(N)$ 이 걸리므로 시간 초과.

BOJ 1280 - 나무 심기

- 절댓값 기호를 풀어보자. X_i 왼쪽 나무와 오른쪽 나무를 구별해서 생각하면
- $cost(i) = \sum_{j < i, X_j < X_i} (X_i - X_j) + \sum_{j < i, X_j > X_i} (X_j - X_i)$
- L 을 X_i 왼쪽에 심은 나무 개수, R 을 X_i 오른쪽에 심은 나무 개수라고 하면
- $cost(i) = (L - R)X_i - \sum_{j < i, X_j < X_i} X_j + \sum_{j < i, X_j > X_i} X_j$
- 즉 X_i 기준으로 **왼쪽, 오른쪽에 심은 각각의 나무의 개수와 좌표의 합**을 빠르게 구해야 한다!

BOJ 1280 - 나무 심기

- 두 개의 Fenwick Tree로 해결할 수 있는데,
- 하나는 나무가 심어질 때마다 해당 좌표에 1을 더하여서 나무 개수를 세고
- 하나는 나무가 심어질 때마다 해당 좌표에 좌표값을 더하여서 좌표의 합을 세는
- 역할을 하면 된다.
- 이를 이용하면 $O(N \log N)$ 의 시간복잡도로 해결 가능.

BOJ 16572 - Pixel Triangles

- solved.ac 기준 난이도: Platinum IV
- 분류: 배열, 누적합

BOJ 16572 - Pixel Triangles

- 하나의 삼각형이 들어올 때마다 업데이트되는 칸의 개수가 매우 많으므로, 다른 방법을 써야할 것 같다.
- 최소한의 정보를 저장해서 삼각형을 나타낼 수 있을까?
- 크기 C 인 픽셀 삼각형은 크기 $C - 1$ 의 픽셀 삼각형 두 개 + 픽셀 한 개로 나타낼 수 있음을 이용!

2	1	1
1	1	0
1	0	0

BOJ 16572 - Pixel Triangles

- A, B, C 가 들어오면, (A, B) 에 C 를 저장하자.
- 모든 정보가 다 들어오면, 이차원 배열을 순회하면서 (x, y) 에서 $(x + 1, y)$ 와 $(x, y + 1)$ 에 $A[x][y] - 1$ 의 값을 넘겨주자.
- 이러한 방식을 반복하면? 픽셀 삼각형에 포함되는 부분은 0보다 큰 값이 저장되어 있게 된다.
- 시간복잡도는 $O(N + 4000000)$

BOJ 13303 - 장애물 경기

- solved.ac 기준 난이도: Platinum IV
- 분류: 정렬, 자료구조

BOJ 13303 - 장애물 경기

- 목적지까지의 x좌표 상의 거리는 모든 경로마다 동일할 것이다.
- y좌표 상의 이동 거리만 최소화하면 됨.
- 장애물마다 오른쪽으로 이동할 때의 y좌표가 두 개
- 두 가지 경우에 대해 모두 백트래킹한다면? TLE.

BOJ 13303 - 장애물 경기

- x 축에 수직하게($=y$ 축과 평행하게) 무한히 긴 직선을 하나 준비하자.
- 시작점에서부터 오른쪽으로 미는데, 장애물을 만난다면 직선 위에 장애물 양 끝점에 해당하는 곳에 거기까지의 최단거리를 표시해두자.
- 이를 도착점까지 유지하고, 마지막에는 최단거리가 가장 짧은 곳들만 출력하면 해결할 수 있음.

BOJ 13303 - 장애물 경기

- 이를 구현하는 방법은?
- 거리를 표시할 수 있는 무한히 긴 직선을 `std::map<int, int>`가 대신할 수 있다.
- map을 들고 다니면서, 장애물을 만날 때마다 영향이 가는 y좌표들에 대해 거리를 매번 갱신해주면 해결할 수 있음.
- 시간복잡도는 $O(N \log N)$

BOJ 2572 - 보드게임

- solved.ac 기준 난이도: Gold V
- 분류: 동적계획법

BOJ 2572 - 보드게임

- 현재 어떤 카드를 내야 하고, 몇 번 마을에 있는지에 대한 최대 점수를 알면 중복 방문을 줄일 수 있다.
- $dp(i, j) := i$ 번 카드를 내야 하고, j 번 마을에 있을 때 최대 점수
- 그렇다면 j 번 마을과 연결된 마을로 이동할 때 얻는 점수도 알 수 있고, 초기 상태와 최종 정답도 $dp(i, j)$ 로 구할 수 있음을 알 수 있다.
- 시간복잡도 $O((N + M)K)$

BOJ 2457 - 공주님의 정원

- solved.ac 기준 난이도: Gold IV
- 분류: 그리디

BOJ 2457 - 공주님의 정원

- 구간 상에 주어진 선분들을 이용해, 주어진 구간을 최소 개수의 선분으로 덮는 문제
- 3월 1일이 포함되는 꽃 중 처음으로 골라야 하는 꽃은?
- 가장 오래까지 살아있는 꽃을 고르는 것이 이득임을 알 수 있음.
- 이후에는 아까 고른 꽃이 죽기 전에 심어져서 가장 오래 사는 꽃을 고르는게 이득이고, 이를 계속 반복하면 답을 구할 수 있음.

BOJ 2457 - 공주님의 정원

- 월일로 입력이 들어와 까다롭다고 생각할 수 있으나, 월일을 하나의 자연수로 합쳐 생각하면 편리하다(8월 31일 - > 831)
- 3월 1일부터 11월 30일까지임에 주의하자.