초음속철도 - 풀이

1 부분문제 1/2 (18점)

어떠한 철도편의 집합이 올바른 철도 개편이라는 것은, i 번 역과 $i+1$ 번 역을 잇는 모든 선로에 대해서 해당 선로를 오가는 철도편이 존재함을 뜻합니다. n이 작기 때문에, 선로를 배열로 표현한 후 철도편마다 마킹하는 것으로 이를 확인할 수 있습니다. 고로, 모든 가능한 2^m 가지의 경우의 수를 완전 탐색한 후, 해당 경우가 올바른 철도 개편인지를 판별하면 됩니다.

부분문제 2는 n이 커서 이러한 방법이 불가능하지만, 그럼에도 방법이 있습니다. 숫자가 크기는 하지만, 실제로는 이렇게 큰 숫자들은 의미가 없습니다. 숫자들 간의 대소 관계만 비교하면 되기 때문입니다. 고로, 좌표 압축이라는 방법을 사용해서, 숫자의 크기를 줄이면 부분문제 1과 똑같은 방법으로 문제를 해결할 수 있습니다. 좌표 압축에 대한 자세한 설명은 생략합니다.

이후 풀이는 모든 입력값이 좌표압축 되어서, $n \le 2m$을 만족한다는 가정 하에 서술을 진행하겠습니다.

2 부분문제 3 (43점)

편의상 모든 철도편이 서로 다르다고 가정하고 서술하겠습니다. 몇가지 정의를 하고 시작하겠습니다.

정의 1 *서로 다른 철도편 i, j에 대해서, 만약 $S_i \le S_j, E_j \le E_i$ 를 만족한다면, 철도편 i는 철도편 j를 지배한다.*

정의 2 *철도편의 집합이 주어졌을 때, 이 집합의* ***뼈대 집합****은, 집합 안에 속하는 다른 철도편에 지배당하는 철도편을 모두 제거한 부분집합을 뜻한다.*

뼈대 집합이라는 이상한 개념을 정의한 이유는, 재미있는 성질이 있기 때문입니다. 먼저, 올바른 철도 개편의 뼈대 집합은, 모든 선로를 정확히 한 번씩만 덮습니다. 이에 대한 증명은 만약 두 번 이상 덮게 될 경우 한 쪽이 지배당하는 일이 발생하고, 0번 덮게 될 경우 올바른 철도 개편이라는 조건에 모순이 되는 식으로 가능합니다.

준비가 다 되었으니, 올바른 철도 개편을 세지 말고, 올바른 철도 개편의 서로 다른 뼈대 집합의 개수를 세어 봅시다. 이는 간단한 동적 계획법으로 가능합니다. $D_i = [1, i]$ 구간의 철도를 덮는 뼈대 집합의 경우의 수 라고 했을 때, 끝점이 i인 철도편을 모두 순회하면서 이 표를 $O(m)$에 채울 수 있습니다.

이를 변형해서 올바른 철도 개편의 개수를 셀 수 있습니다. 골자는 어떠한 부분 집합을 뼈대 집합으로 가지는 올바른 철도 개편의 개수를 각각 세어서, 모두 더하는 것입니다. 동적 계획법 과정에서 뼈대 집합에 철도편을 하나 넣게 되면, 해당 철도편에 지배당하는 다른 철도편들은, 자유롭게 넣거나 넣지 않을 수 있습니다. $C_i = i$번 철도편에 지배당하는 철도편의 개수 를 셌다고 했을 때, $D_i = \sum_{E_j=i} D_{S_j} \times 2^{C_j}$ 라는 식이 유도됩니다.

마지막 과제는 C_i를 계산하는 것입니다. 일반적인 경우에 C_i를 계산하는 것은 자료 구조를 수반하지만, 부분문제 3의 조건을 사용하면, 스택을 사용해서 C_i를 선형 시간에 계산할 수도 있습니다. 결론적으로 정렬과 좌표 압축을 제외하면 모든 과정을 선형 시간에 처리할 수 있습니다.

앞에서 설명했던 것과 다르게 실제로 모든 철도편이 서로 다르지 않으며, 중복된 철도편 역시 존재합니다. 이 부분을 처리하는 것은 알고리즘적으로 어려운 게 아니라 단순히 코딩이 더 복잡해 질 뿐이니 설명을 생략하겠습니다. 전체 시간 복잡도는 $O(m \lg m)$ 입니다.

3 부분문제 4 (45점)

부분문제 3을 풀기 위해서 뭔가 열심히 끄적거렸지만, 애석하게도 부분문제 3의 풀이는 만점 풀이와 아무 상관이 없습니다 ㅠㅠ 조금 더 일반화된 풀이를 생각해 봅시다.

$D_{i,j}$ = 1번 구간에서 i번 구간까지를 모두 처리했으며, 그 과정에서 $[1, j]$ 구간을 모두 덮는 경우의 수라고 정의합시다. i 번 구간을 넣었을 때의 결과를 계산하게 되면, 다음과 같은 상태 전이가 가능합니다.

- $j < E_i$ 일 경우, $D_{i,j} = D_{i-1,j}$
- $j = E_i$ 일 경우, $D_{i,j} = D_{i-1,j} + \sum_{k=S_i}^{E_i} D_{i-1,k}$
- $j > E_i$ 일 경우, $D_{i,j} = D_{i-1,j} \times 2$

고로 $O(m^2)$에 동적 계획법 표를 모두 완성할 수 있습니다.

4 부분문제 5 (100점)

부분문제 4의 동적 계획법 표가 $O(m^2)$ 크기이기 때문에 섣불리 줄이기 힘들 것이라고 생각할 수도 있겠지만, 상태 전이가 상당히 깔끔하기 때문에 표의 크기를 줄이려는 추가적인 노력 없이 만점 풀이를 얻을 수 있습니다.

동적 계획법의 상태 전이를 관찰해 보면, 지금 수행하고 있는 연산은 아주 간단합니다. $D_{i,*}$ 라는 배열이 있을 때, 우리는 1. 구간 합을 구하거나 2. 구간에 2배를 곱해주거나 3. 원소 하나를 갱신해주거나 와 같은 3가지 연산을 총 $O(m)$번 수행해서, 표를 완성하는 꼴이 됩니다.

고로 저 세가지 연산을 빠르게 수행할 수 있다면, 만점을 얻을 수 있습니다. $D_{i,*}$ 를 세그먼트 트리라는 자료 구조의 형태로 관리하는 것이 핵심으로, 1번 연산과 3번 연산이 있을 때는, 단순히 트리를 타고 내려가면서 값을 봐주면 되고, 2번 연산은, 구간에 대한 "Lazy propagation" 방법을 사용해서 효율적으로 처리할 수 있습니다. 세그먼트 트리와, "Lazy propagation" 방법에 대해서는 자세한 설명을 생략하겠습니다.

"Lazy propagation"을 사용한 세그먼트 트리를 통해서, $O(\log m)$에 각각의 질의를 처리할 수 있으며, 고로 총 시간 복잡도 $O(m \log m)$에 모든 문제를 해결할 수 있습니다.