"외곽 순환 도로 2 (ringroad2)" 문제 풀이

작성자: 구재현

부분문제 1

모든 홀수 사이클과 교집합이 있는 간선 집합이라는 것은, 해당 간선 집합을 제거하였을 때 남은 그래프가 이분 그래프 (Bipartite Graph) 라는 것이다. 이분 그래프의 경우 2-Coloring이 존재한다. 다시 말해, 모든 간선의 양 끝점이 서로 다르도록 정점의 색을 **흰색** 과 **검은색** 으로 칠할 수 있다는 것이다.

따라서, 간선 집합을 열거하는 대신, 정점의 색을 모두 열거한 후 **같은 색을 잇는 간선의 가중치 합** 을 최소화하는 식으로 문제를 변형할 수 있다. 이렇게 변형한다고 문제가 크게 달라지지는 않지만, 이 구조가 DP를 적용하기 가장 적절하다.

서브태스크 1의 경우, 외곽 순환 도로를 이루는 정점의 수가 작다. 이 정점들의 색의 경우는 2^K 가지이다. 이 경우의 수를 모두 열거하자. 이 경우 외곽 순환 도로들의 양 끝점 색이 고정되니, 가중치를 더하거나 더하지 않은 후 제거할 수 있다. 남은 그래프는 트리가 된다.

이제 다음 문제를 풀면 된다:

- 트리가 주어지고, 일부 정점에 색이 고정되어 있을 때, 모든 정점에 흰색/검은색 을 부여해서, 같은 색을 잇는 간선의 가중치 합을 최소화하라.

이는 $DP[v][c] = (v$ 의 서브트리에 모든 색을 칠했고, v 의 색은 c 일 때, 서브트리에서 필요한 비용) 이라는 DP를 정의하면 해결할 수 있다. 상태 전이는 간단하다.

트리 DP를 선형 시간에 해결할 수 있고, 이를 2^K 번 해결하니, 시간 복잡도는 $O(2^K N)$ 이다.

부분문제 2

부분 문제 1의 관찰을 사용하자. 일반성을 잃지 않고 0 번 점이 흰색이라고 하면, 사이클 상 색의 모든 조합을 DP로 열거한 후 이에 따른 비용을 계산할 수 있다. 부분 문제 1의 관찰이 없이도 유사한 DP를 얻을 수 있으나 정의나 상태 전이가 더 복잡할 것이다. 시간 복잡도는 $O(N)$ 이다.

부분문제 3

외곽 순환 도로를 끊을 수 없기 때문에, 순환 도로 상 리프의 색이 모두 고정된다. 부분 문제 1과 동일하게 해결하면 된다. 시간 복잡도는 $O(N)$ 이다.

부분문제 4

부분 문제 3과 다른 점은 K 가 홀수일 경우 외곽 순환 도로 중 하나를 끊어야 한다는 것이다. 리프의 색상 조합이 $O(N)$ 개 생기는데, 이는 모두 열거해 보기엔 많다. 부분 문제 3의 DP를 수정하여, 서브트리가 끊겨진 외곽 순환 도로의 왼쪽에 있는지, 오른쪽에 있는지, 혹은 그 사이인지를 인자로 저장한다. 리프의 색은 외곽 순환 도로 상 위치, 그리고 위 인자에 따라 결정된다. 이 경우 모든 리프의 색상 조합을 열거함과 동시에 최적해를 구할 수 있다. 시간 복잡도는 $O(N)$ 이다.

부분문제 6

부분문제 1과 비슷하게 트리 DP로 해결하는데, 외곽 순환 도로에 대한 고려가 필요하다. 이를 고려해 주기 위해서, 다음과 같이 DP 상태를 정의하자:

- $DP[v][c][l][r]$ = (v 의 서브트리에 모든 색을 칠했고, v 의 색은 c 이고, v 의 서브트리에서 *왼쪽으로 나가는* 외곽 순환 도로와 인접한 정점의 색이 l, *오른쪽으로 나가는* 외곽 순환 도로와 인접한 색이 r 이다.

0번 정점을 루트로 해서 트리를 구성하면, 서브트리 안과 밖을 잇는 외곽 순환 도로는 정확히 두 개가 될 것이다. 이 두 외곽 순환 도로의 서브트리 안쪽 정점 색을 저장한다고 보면 된다.

각 자식에 대한 DP 값을 재귀적으로 계산한 이후, 상태 전이를 진행한다. 두 서브트리를 합쳐줄 때, 두 서브트리 사이를 잇는 외곽 순환 도로가 다른 색의 정점을 잇는지, 같은 색의 도로를 잇는지에 따라서 가중치가 추가될 수도 있고 아닐 수도 있다. 예를 들어, $DP[v_1][c][l_1][r_1]$ 와 $DP[v_2][c][l_2][r_2]$ 를 합친다고 하면, $r_1 = l_2$ 일 경우 비용을 지불해야 할 것이고, 합친 결과는 $DP[v][c][l_1][r_2]$ 와 같을 것이다.

이러한 식으로 상태 전이를 고려해 주면, 각 정점당 $O(1)$ 시간에 DP 테이블을 합칠 수 있다.

매 합치는 과정에서 외곽 순환 도로의 비용을 찾는 것이 어려울 수 있는데, 만약 위 과정을 일반적인 DFS로 하게 된다면, DFS를 하는 과정에서 외곽 순환 도로의 비용을 preorder 순으로 읽기 때문에, 단순히 큐와 비슷하게 필요할 때 맨 앞에 있는 Weight를 뺄 수 있다. 이러한 방식으로 구현하면 코드가 아주 짧으나, 그 외 다른 방식으로 구현하여도 크게 어렵지 않다.

시간 복잡도는 $O(N)$ 이다.

여담

위와 같은 문제를 Odd-Cycle (Edge) Transversal 이라고 하며, 2020 선발고사의 *아이싱* 문제도 이러한 류의 문제에 속한다. Treewidth가 $tw(G)$ 인 그래프에 대해서, Odd-Cycle Edge Transversal은 $O(4^{tw(G)}N)$ 에 트리 DP로 해결되니, Odd-Cycle Edge Transversal은 Treewidth에 Fixed-Parameter Tractable하다.

이 문제에서 다룬 그래프는 $tw(G) \leq 3$ 이기 때문에 위 풀이를 적용할 수 있으며, 실제로 이러한 코드로도 만점을 받을 수 있다. 위에서 다룬 풀이는 그래프의 Ad-hoc한 성질을 더 활용하여 간소화한 풀이라고 할 수 있다.