"넘버링 (numbering)" 문제 풀이

작성자: 구재현

어떠한 수열의 *비다양성* 을 $\sum_{0\le u,v\le n-1} \mathbb{I}[A[u]=A[v]]$ 로 정의하자. (비다양성) + (다양성) + (다양성) $= N^2$ 이기 때문에, 비다양성의 최솟값을 구하면 다양성의 최댓값을 쉽게 구할 수 있다. 아래 풀이에서는 비다양성을 최소화하는 문제를 해결한다.

또한, 굿 넘버링에 등장하는 서로 다른 숫자가 T 개일 때, $A[i] \in \{0,1,\ldots,T-1\}$ 임을 가정한다.

부분문제 4

문제의 정의에 따라서, $A[u]=A[v]$ 인 서로 다른 두 정점 u,v 를 잇는 경로에는 같은 숫자가 적혀 있어야 한다. 이에 따라, $A[u]$ 가 같은 정점들은 트리에서 서로 연결되어 있다. 이 정점들을 하나로 압축할 경우, 트리는 경로가 된다: 압축된 트리에서 차수 3 이상의 정점이 있다면, 이 정점보다 $A[v]$ 값이 작은 정점이 2 개 있거나 큰 정점이 2 개 있고, 그들간의 경로가 문제 조건을 만족하지 않는다.

최적해에서 $A[x]=0, A[y]=T-1$ 인 두 정점을 고정하자. 위 관찰에 의해서, x 와 y 간의 경로에 속하지 않는 트리 간선 (u,v) 는 $A[u]=A[v]$ 를 항상 만족한다. x 와 y 간의 경로에 있는 간선들을 지우면, 트리는 몇 개의 컴포넌트로 분할되고, 각 컴포넌트에는 항상 같은 색이 배정되어야 한다. 이 경우, 경로를 따라가면서 각 컴포넌트마다 값이 증가하도록 다른 색을 배정해 주면 현재 조건에서 비다양성을 최소화하는 굿 넘버링을 얻을 수 있다. 고로, 이 경우 얻을 수 있는 비다양성의 최솟값은 각 컴포넌트 크기의 제곱의 합이다.

모든 가능한 x,y 에 대해서 위 절차를 구현해 주고, 비다양성의 최솟값을 계산해 주면 $O(n^3)$ 에 전체 문제를 해결할 수 있다.

부분문제 5

최적해에서 x,y 가 트리의 리프임을 가정할 수 있다. x 를 고정하고, 모든 y 에 대한 답을 DP로 해결하자. 다음과 같은 정의를 도입한다:

- $child[v]$: v 의 자식 집합
- $sub[v]$: v 의 서브트리 크기
- $DP[v]$: y 가 v 의 서브트리에 있다는 가정 하에 v 의 서브트리를 컴포넌트로 나눴을 때 비다양성의 최솟값

이 때, $DP[v]=\min_{w\in child[v]}(DP[w]+(sub[v]-sub[w])^2))$ 가 성립한다. x 를 고정했을 때 이 점화식은 $O(n)$ 시간에 계산할 수 있으니, 전체 문제는 $O(n^2)$ 시간에 해결할 수 있다.

부분문제 3

임의의 정점 (편의상 0 번 노드) 를 루트로 고정하고 DP를 계산하자. 정점 l 에 대해서, $LCA(x,y)=l$ 인 모든 (x,y) 에 대한 비다양성의 최솟값은, $\min_{u,v\in child[l],u\ne v}(DP[u]+DP[v]+(n-sub[u]-sub[v])^2)$ 와 같이 계산할 수 있다. 이를 모든 정점 l 에 대해서 계산해 주면, DP를 매번 다시 계산할 필요 없이 문제를 해결할 수 있다. 시간 복잡도는 모든 정점의 차수 합의 제곱으로, 이진 트리에서는 $O(n)$ 시간이지만 일반 트리에서는 여전히 $O(n^2)$ 이다.

부분문제 6

위 식을 풀어쓸 경우, $\min_{u<v,u,v\in child[l]}(DP[u]+DP[v]+(n-sub[u])^2+sub[v]^2-2(n-sub[u])sub[v])$ 와 같이 전개된다. 고정된 v 에 대해서, 이는 일차 함수의 최솟값 쿼리와 동일하다. 필요한 일차 함수들을 기울기 순으로 정렬한 후 Convex Hull Trick을 사용하면, 전체 문제를 $O(n\log n)$ 에 해결할 수 있다. 출제진이 *의도한 풀이*는 이것이었고, 실제 대부분의 학생들이 이렇게 해결하였다.

l 의 자식들을 $(sub[u], DP[u])$ 값을 기준으로 정렬하자. $sub[u]$ 가 같은 값이 3 개 이상 있다면, 이 중 $DP[u]$ 가 가장 작은 두 값을 제외하고는 모두 계산에서 무시해 주어도 무방하다. 그러한 값을 사용한다면 가장 작은 두 값으로 대체하는 것이 항상 이득이기 때문이다. 필요 없는 자식들을 제거한 후, 단순히 모든 자식 쌍을 순회할 경우 현재 입력 데이터에서 만점이 나온다. 코드가 짧기 때문에 대부분의 출제진들은 이렇게 해결하였다.

서로 다른 $sub[u]$ 값이 최대 $O(\sqrt{n})$ 개이기 때문에, 이 방법이 $O(n\sqrt{n})$ 의 시간 복잡도를 가지는 것은 증명할 수 있다. 하지만, 이보다 더 강한 증명은 알지 못한다. 출제진은 동적 계획법으로 모든 트리 중 (서로 다른 sub[u] 값의 수)^2 의 가능한 최대 합을 계산해 보았고, $n=10000$ 에서 대략 5만, $n=100000$ 에서 대략 60만 이하의 값이 계산되었다. 사용한 동적 계획법이 $O(n\sqrt{n})$ 의 메모리를 사용해서 $n=1000000$ 에서의 최댓값을 산출하지는 못하였지만, 추세를 보았을 때는 의미 있는 반례가 존재하지 않을 것이라고 추측된다.

부분문제 9

어떠한 서로 다른 두 정점 u,v 에 대해서 u,v 를 모두 포함하는 사이클이 있다면, $A[u]=A[v]$ 여야 한다. 즉, 같은 이중 연결 요소에 속하는 노드들은 같은 값을 배정받아야 한다. 이중 연결 요소를 하나의 정점으로 합쳐주면, 그래프는 트리가 된다. 각 트리 정점이 원래 그래프의 여러 정점에 대응될 수 있다는 차이 외에는, 서브태스크 6과 동일한 풀이를 사용하여 해결할 수 있다. 이중 연결 요소는, 그래프의 절선을 구한 후, 절선이 아닌 간선들의 양 끝점을 (Union-Find 등으로) 합쳐주는 식으로 계산할 수 있다. 시간 복잡도는 (CHT 사용 기준) $O(m+n\log n)$ 이다.