



猫か犬 (解説)

問題をグラフの言葉で言い換えると、庭の構造は木構造であり、各小屋は頂点と言い換えることができる。簡単のため、猫がいる小屋を「赤」の頂点、犬がいる小屋を「青」の頂点と呼ぶことにする。

小課題 1

木の頂点の個数が少ないので、危険度を求める際に、どの辺を切断するかすべて試すことができる。辺を切断するかどうか決めると判定するのは簡単なので、 $O(N \cdot 2^{N-1})$ の時間計算量で各クエリごとに解くことができる。

小課題 2

各クエリに対して、線形的时间計算量で解くことができればよい。先ほどはグラフの構造を利用していなかったもので、木を用いることを考えていこう。

そこで、木 DP することを考えよう。適当な頂点を根として根付き木として考える。各頂点 v について、 v を「赤色の連結成分」の属する頂点にするときに、切断する必要がある辺の最小本数を R_v とする。また、 v を「青色の連結成分」の属する頂点にするときに、切断する必要がある辺の最小本数を B_v とする。これを葉から順に更新していくことを考える。ここで、「赤色の連結成分」とは、辺を切断したときに青色の頂点が存在しない連結成分のことを指し、同様に、「青色の連結成分」も定義する。

このとき、漸化式は以下ようになる。

$$\begin{aligned} \bullet B_v &= \sum_{c: \text{a child of } v} \min(B_c, R_c + 1) \\ \bullet R_v &= \sum_{c: \text{a child of } v} \min(R_c, B_c + 1) \end{aligned}$$

これを DFS で順に計算していけばよい。よって、各クエリ $O(N)$ で解くことができるので、全体で $O(NQ)$ の時間計算量で解くことができる。

小課題 3 (満点解法)

先ほどの解法で利用していなかった性質を考えよう。それは、頂点の状態が高々 1 頂点でしか変わらないということだ。

そこで、先の漸化式を高速化することを考えると、Heavy Light Decomposition が思い当たるだろう。このテクニックについての解説はネットを探せばたくさん見つかると思うので、そちらに任せる。今回は、実際に HLD をした際に、どのようなデータ構造を乗せるべきかについてのみ述べる。



それは元のグラフの構造が列の場合にどのように解くか、ということに帰着できる。しかし、そのまま列の場合について考えてしまうと、簡単に解けてしまう。そこで、各頂点について、 B_v, R_v があらかじめ定まっている状況を考えれば、正確な帰着になっていそうである。そこで、そのように帰着すると、以下のようなになる。

- 一列に N 個のマスが並んでおり、各マスを赤か青に塗る。
- 各マス i について、赤なら R_i 、青なら B_i のコストが付く。
- ただし、色の異なるマスが隣接していた場合、そのマスの組に対して、1 のコストが付く。
- 全体のコストを最小化せよ。

このように書くと、Segment Tree で解くことができそうに思えるだろう。実際、各区間について、左右を赤/青のどちらにするかを決めたときに、全体のコストの最小値を持っておけばよい。

同様にすることで、各区間について $O(\log N)$ で更新できるから、全体としてクエリあたり $O(\log^2 N)$ の時間計算量で解くことができる。