

KUPC2020 spring C: サボテンクエリ 解説

writer: kazuma

2020 年 3 月 20 日

解法

まず木の問題に落としてみましょう。各単純閉路に頂点をひとつ追加して、閉路上の辺を削除して、追加した頂点と元の閉路上の頂点とをつなぐ辺をそれぞれ追加します。すると、カクタスグラフ*¹はこの操作で木になります。これは連結性が保たれていて、閉路がなくなっていることから示せます。

さらに、追加された頂点に元となった閉路の全ての辺のコストを XOR したコストを持たせます。それ以外の頂点は 0 を持たせます。

すると、元のグラフで x - y 間の単純パスのコストを一つ D としたとき、木に変形したグラフ上で x - y パス上にある頂点のコストをコスト D に対して自由に XOR して作れるコスト集合の中で k 番目の値を求める問題になります。

単純パスのコストを一つ求めるのは、元のグラフの全域木を一つ作ってそこでの x - y パス上の XOR コストを計算することで得られます。あとは、カクタスグラフから先ほどの手順で変形した木上で、 x - y パス上の頂点のコストの XOR の基底を高速に求めることができればクエリ処理できます。

ここで、HL 分解とセグメントツリーを使うと求められるように思うかもしれませんが、基底のマージ回数が各クエリ $O((\log N)^2)$ 回になって、さらに基底のマージにはビット数を B としたときに (今回の問題では $B = 30$) $O(B^2)$ かかるので重過ぎて TLE します。

そこで、重心分解で解きます。各クエリがどの再帰時点で計算すべきかを求めておけば、基底のマージ回数が各クエリ一回だけで済みます。前処理は木を重心分解して各頂点に再帰段階のランク付けをしてからパス上の最小値を求めることでできます。これはオーダー上この問題のボトルネックにはなりません。

こうして、全体で $O(N \log N \times B + Q \times B^2)$ で求まります。

*¹ 問題文で連結なサボテングラフと呼んでいたものは一般的にカクタスグラフ (Cactus graph) と呼ばれています。知らなかった人はぜひ調べてみてください。