Designated Cities

maroon

問題概要

- N頂点の木がある
- 各辺はそれぞれの向きにコストが定まっている
- 一個頂点を選ぶと、各辺について、選んだ頂点に向かう向きのコストが0になる
- 何頂点か選んでコストの合計の最小化せよ、という問題にたくさん答える

小課題1

- N<=16

- 2^N通り選ぶ頂点を全探索してDFSする
- O(N^2 2^N)

小課題2

- Q=1, 選ぶ頂点は1個

- とりあえず頂点0を選んだ際のコストの総和を求める
- 頂点vでのコスト総和がわかっているとき、vに隣接する頂点uでのコストの総和は、vとの差分を考えると、辺e=(u,v)のコストからわかる

- よってdfsしていけば全頂点について求められる
- O(N)

小課題3

- Q=1, 選ぶ頂点は2個
- 2頂点選んだときのコストの総和は、(それぞれの頂点を独立に選んだときのコストの総和+2頂点間のパスの重み（双方向）の総和)/2という形になっている
- 木の直径を求めるDPと同じ要領でDPできる
- O(N)

小課題4

- N<=2000

- 選ぶ頂点が1個の場合はできるのでそれ以外の場合を考える
- 木の葉の個数以上選べるときは、葉を全部選べばコスト0
- 木の葉の個数未満しか選べないときは、選ぶ頂点は全て葉である（葉でないものを選んだ場合は適当な葉に向けて動かすと得できる）

小課題4

- 選ぶ葉を1個固定する、これはO(N)通り

- 選んだ頂点を根にして考える
- 根に向かう頂点のコストはすべて0になっている
- 辺のコストは葉に向かうものだけ考えればよい
- 葉を一つ指定すると、根からその葉へ向かうパスのコストが0になる
- 葉をいくつか選んで、(根ー葉)を結ぶパス上の辺のコストの総和を最大化する問題になる

小課題4

- 実は、選ぶ葉は貪欲に選んでよい（次に選ぶと増えるコストが最大、というものを選んでいく）

- 証明は簡単
- 最初の段階でコスト最大の(根ー葉)パスを任意にとり、その葉をLとおく
- このとき、Lを使わない解があるとすると、Lを使うように変形して損しないことがわかる

小課題4

- 次に選ぶと増えるコストが最大、というものを選ぶ

- やり方は主に2つ

- SegTreeで頑張る
- それぞれの葉について使ったときに増えるコストをSegTreeで管理する
- あるパスを使ったときの、それぞれの葉における増加コストの変化は、範囲加算等で表現できる

小課題4

- priotity_queueで頑張る
- まずpriority_queueに、最大コストのパスをpush
- 以下を繰り返す
- priority_queueから最大コストのパスを取り出して、使う　　パスを使うことによって、木がいくつかの部分木に分かれることになる
- 分かれる各部分木の最大パスをpriority_queueにpushする
- こっちのほうが簡単

小課題4

- どちらの方法でもO(NlogN)でできる
- 決め打つ葉をO(N)通り試すので、合計O(N^2logN)

小課題5

- これいる？
- 僕はいらないと思います
- 満点解法に繋がらないし面倒なだけなので省略

- 一応簡単に書くと、先の解法でpriority_queueにpushされるコストの集合を考えて、根として選ぶ頂点を（葉に限定しないで）一つ隣の頂点に移動した場合の変化を見る
- 変化の回数が1回の移動あたり定数回なので解ける

小課題6

- 葉を全部試すのが大変
- 実は、2個選ぶときの解で選ぶ頂点のうち一つを適当にとってくる(Rとする)と、2個以上選ぶときの解であってRを含むものが必ず存在

小課題6

- 証明は帰納法を回す
- K個選ぶ解であってRを含むものが存在すると仮定
- K=2は自明
- K+1個選ぶ解を任意にとる
- ある解に対して、選んだ2頂点間のパス上に存在する点を内点と呼ぶことにする
- K個選ぶ解とK+1個選ぶ解で、内点が共有されている場合、そこを起点に先程の貪欲を走らせれば同じ解が得られるはず

小課題6

- 内点が共有されていない場合、K+1個の解で選んだ頂点のうち適切なものを取ると、それをK個の解に付け足すとK+1個の解と比べて損しない解が得られる

- よって示された

小課題6

- 根として固定する葉が1つでいいので、O(NlogN)で通る

得点分布

- 0点　　1人
- 6点　　3人
- 7点　　1人
- 13点　8人
- 16点　1人
- 22点　1人
- 30点　1人
- 39点　4人
- 100点 1人