

2018年春合宿 Day2 Road Service

チューター 劉

問題概要

- N頂点の木が与えられる
- K本の辺を新たに追加し、総距離を最小化
 - 総距離=2点間の最短距離の総和
- $N=1000$
- OutputOnly

問題概要

- N 頂点の木が与えられる
- K 本の辺を新たに追加し，総距離を最小化
 - 総距離=2点間の最短距離の総和
- $N=1000$

• **OutputOnly !**

OutputOnly

- 入力を見て作業できる
 - 特徴が分かればその入力に特化してもよい
 - ビジュアライザもしばしば有効
 - 手で解いてしまってもよい
- 厳密解を出せなくてもよい
 - だいたい上手いきそうな貪欲・DPで高いスコアが出せることも
 - 初期状態を決めて山登り

OutputOnly

- 5時間中に終わられることが重要
 - さらに，5時間掛けるわけにもいかない
 - 終わりがないので最後に回す？
 - バックグラウンドで回すために序盤にやる？
 - なるべく短い時間でほどほどの得点を出し，残りの問題に時間をかけたい
 - 楽な方針をまずは探そう
- 何度も提出できる
 - 細かいパラメータ変更を行えるよう設計したい

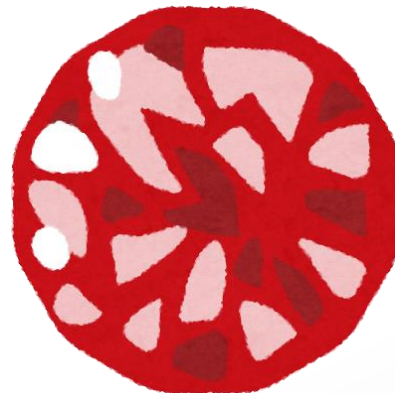
OutputOnly

- その他，細かいこと

- (今のパソコンはマルチコアなので) 例えば端末を複数開き同時実行
- 実行時間制限はないので，山登りや探索をひたすら繰り返し，解が更新されるたび出力
- 標準出力のバッファがあふれることも→リダイレクトしてファイルを開いて切り貼り

```
./a.out < in > out
```

- C++ではなくスクリプト言語を用いてもよい



方針

- 最小化したいのは全点間の最短距離の総和
- 全点間の最短距離の総和は，木について
 - 直線だと N^3 オーダー
 - ウニだと N^2 オーダー
- 1つの頂点に集中していると強い
 - "ハブ空港"を作るイメージ

方針

- 1つの頂点に集中していると強い
 - "ハブ空港"を作るイメージ
- 実は，無作為からの山登りも長い間続けると自然とハブを作りたがる
 - ただし，単純山登りしているとハブを作るまでが長すぎて難しい
- 単に頂点1-2, 1-3, ... 1-(K+1) と出力するだけで30点付近



方針

- ハブを決めたとして，どの頂点とハブを直接つなぐか？
- 総合的にどの頂点もハブに近いようにしたい
 - 単純な戦略の1つ→次数の高い頂点につなぐ
 - 他にも工夫の余地アリ
- ハブをN通り試し，最も良かったものを出力
- だいたい60点



方針

- ハブとつなぐ頂点の集合 X を，単に次数の大きい頂点にするのはさすがに雑
- この状態から山登りをしよう
- 山登りの一例:
 - X から1点取り除き，1点追加する操作を1ステップで行ける近傍とする
 - 近傍のうち最もスコアのいいものに遷移
 - 近傍は $O(NK)$ 通り



方針

- 山登りで，評価関数を総距離とすると遅い; $O(N^2)$
- ハブから他の頂点への距離の総和で代用; $O(N)$
 - 実際に，すべての頂点对がハブを経由するなら大小を保つ
- これで近傍の広さと山登り回数が稼げる
- 初期状態を全ハブ試すのはやめて，その時点での評価トップ10ぐらいに対して山登る→だいたい100点



方針

- 次数の高い頂点につなぐ代わりに「最も総距離を減らす頂点に向けてハブから辺を貼る」をK回繰り返す
 - 山登りなしでだいたい90点
 - 総距離の計算は先述の評価関数で近似



得点分布

