2014/03/02 @ LINE 株式会社

東京大学プログラミングコンテスト 2013 (年度)

問題 J : K 番目の閉路

問題：秋葉 (@iwiwi)
解答：秋葉，矢野
解説：秋葉

ナイーブな解法 (TLE)：最良優先探索をする

最良優先探索 (≒Dijkstra) をする

- ただし，１つの頂点を**複数回訪問**する
- i 回目の訪問が i 番目の経路の長さ
- $O(k(n+m)\log n)$ 時間．TLE.

想定解法：A* 探索をする (だけ！)

最良優先探索の代わりに A* 探索を行う

- 頂点 0 **まで**の距離を全頂点に求める
  - 逆グラフで 1 回 Dijkstra するだけ
- それを距離の下界として A* 探索
- A* 探索にするだけで計算量が本質的に改善することを今から説明します

準備

- 頂点 0 **まで**の最短路木を考える
  - 木に含まれる辺を「順路」,
  - それ以外を「迂回路」と呼ぶ

考察 1

経路たちはツリーを成す

- 一番短い経路は順路のみを通る最短路
- i 番目 ($i > 1$) に短い経路には「親」がある
  - ある j 番目の経路 ($j < i$) が存在して
  - 途中まで完全一致して
  - 途中で違う「巡回路」を通って
  - そこからは「順路」のみを通ってゴールする

考察 2

ランダム生成されたグラフの性質

- 全頂点の次数がほぼ均一
  - だいたい 4（全 10^6 通りの入力での最大次数でも 23）
- 最短路木の深さが浅い
  - 全ての 10^6 通りの入力で平均 22.2, 最大 35
  - 乱数でこんな単純に作られたグラフはそんな細長くなったりしなそうですよね

※ちなみに，乱数で単純にグラフを作ってしまうとこういう扱いやすい性質を持ってしまうので，一般のグラフを対象とする問題を出題するためのジャッジデータを作る時はもっと変なグラフを生成するようにしましょう

まとめると

最短路木の深さを s，平均次数を d とする

（次数のばらつきはかなり小さいので以下では均一とみなして解析）

i 番目までの経路が求まっているとき

- j 番目 $(j \leq i)$ の経路を親とする全ての経路はもう順路を進むだけで求まる位置で順位キューに入っている
- 順路しか進まない場合の経路の長さは既に順位キュー内でのキーに完全に一致
- なのでまっすぐ進んで，$O(sd \log(ksd))$ で次の経路が求まる

全体 $O\big((n+m)\log n + ksd \log(ksd)\big)$ 時間

（A* 探索で本質的に計算量が向上した！）

別解：グラフを変換して最良優先探索

グラフを変換

- 順路と迂回路に分ける
- 頂点 0 までの距離を使ってグラフを変換
  - ポテンシャルを使う最小費用流と同様に
- 順路が距離 0 になる

最良優先探索

順路は特別扱いとして，順路を進むことはキューに追加せずそのまま処理する

別解というかこっちを先に考えて準備していて A* は後で思いついた．こっちのほうが高速.

結果

- First Accept
  - hirosegolf (209:55)

- 正解者
  - 3 人 (hirosegolf, ir5, cos)