日本橋ハーフマラソン 2024夏 (AtCoder Heuristic Contest 036)

tomerun

感想

- 上位の解法がここまでばらけるのは珍しい、面白い
- コンテスト開始前に書いていた回答は85位相当くらい。伸び代はあると思っていたがここまでとは
- けどトップでもまだ真の最適解とは差がある状態だと思う、実行時間無制限でオフライン実行の問題としてやるのも面白いかも

問題の元ネタ

- スパコンのベンチマークでGraph500というのがある。数兆頂点、数十兆辺という規模のグラフでBFSをするというもの
- 一つの計算ノードのメモリにはグラフの情報が収まらないので、複数のノードで分割して並列に計算し、境界については他のノードと通信して情報をやり取りしたりする
- そこでソフトウェア的な工夫をする必要がある
  https://www.riken.jp/pr/news/2024/20240513_2/index.html

(1)複数のノード間におけるグラフデータの効率的な分割および圧縮
(2)冗長なグラフ探索を削減するアルゴリズム
(3)BFSの結果に影響を与えずに不要な頂点を削除する前処理
(4)スーパーコンピュータの大規模ネットワークにおける通信性能の最適化
(5)探索アルゴリズムの動作を制御するパラメータの自動チューニング

また、他のノードと通信してデータをとってくるようなレイテンシーの高い処理は、通信するデータ量よりも何回通信したかの回数だけが重要で、できるだけデータをまとめて取れるようにするなどしてその回数をどう減らすかが課題になる

こういったテーマで問題を作れないかと考えていた

なぜこの問題を採用したか

- 良い解法がどういうものか、軽く考えるだけでは全然わからないが、確かに強い方法は存在してそう感が高かったので。
- また、配列Aはどのようなものが良いのかについて、(”メタ”ではない)ヒューリスティックをたくさん盛り込む余地があって、思考力と実装力が問われそうだったので。
- 探索パートの重要度がそこまで高くなく、重要なのは天才かつ筋肉構築の部分なので、非インタラクティブ問題の中では比較的速くない言語でも上位が狙いやすい問題だったのでは。

作問過程での問題内容の変化①

最初は

- LA=2N(固定)
- LB=4 ~ 100

という今思えばゆるゆるな制約だった。

LBが大き目の場合は高速道路のような道を作って固定で青にして、そこから脇道を生やすような回答になるのかと思っていた

簡単な回答を書いてみると、LBが大きいと特に工夫せずとも何をやっても1ターン1手以内で行けてしまう感じだったので上限を減らした

作問過程での問題内容の変化②

1回以下の信号操作ではざっくり LB^2 箇所に移動できるわけで、そういうことからLBの最大値24がだいたい √N くらいになっている

LAも、小さいケースもあった方がどうやって詰め込むかを工夫する余地が広がりそうだったので減らした

信号操作に関して。当初はA->Bだけではなく、A->A, B->B, B->A どこへでもコピーできるルールだった。そしてAの初期値は自由ではなく [0,1,2,...,N-1,-1,-1,...,-1] 固定だった

作問過程での問題内容の変化③

良さげな経路をB上に作りながらAに書き戻していって再利用する、といった解法を想像していたが…

考えてみたら経路をB上に構築してAに書き戻すのと、A上に構築してBにコピーして使うのと信号操作の回数は同じなので、結局最初にA上に好きな列を構築するパートが入るだけになりそう。

そしてそのパートは特に面白くなさそうなので、最初から好きに決められるようにして、コピー対象もA->Bだけに単純化した。

サンプルコードについて

- 最初は幅優先探索で最短経路をたどる解法がサンプルだったが、そこから改善するのが結構難しいのでわざと弱くした
- 頂点の座標情報は解法に使用する想定はあまりなかったのですが、結構使っていた人もいそう…？

不要な頂点を無視する①

- 実は600個の頂点のうち200個くらいは一度も目的地として登場しない。それらの頂点を無視することでグラフを小さくできる
- だが全部無視するとたいていグラフが非連結になって回答不可能になる。かといって連結になるような最低限だけ残すとしても疎なグラフになってスコアが落ちる（解法によってはそうでもないかもしれない）
- どの頂点を取り除くかは、取り除いても他の頂点ペアの最短経路長が変化しないようなら取り除く、といったルールベースでやっても良いが…

不要な頂点を無視する②

- 削除する頂点を決めるのに旅行の経路の情報を使うことができる。連続する目的地間の最短経路を一つずつとって、それらに一度も登場しない頂点は取り除く、としたらけっこう良かった
- さらにそれらの経路を一つずつ順に、いったん消してから他の経路で使っている辺とできるだけ共有するような経路を再構築する、を繰り返して、旅行でたどる辺を集約していくことでさらに改善できる
- 平均して頂点100個くらい削れた

最大LB個ずつの連結な頂点のグループにまとめる解法

- 300位あたりで多そうな解法
- 1グループ分まとめて信号をONにすることで何頂点かまとめて移動できる
- グループの作り方でスコアが変わる。できるだけグループ間をまたぐ回数が少なくなるような選び方が良い
- 究極的には、どのグループ同士も直接移動できる形が最高
- グループ間で頂点が重複しないようにとっているとまず不可能だが、LA>＝N であることを利用して一部の頂点を共有するようにすれば、できる

中央から伸ばしてどの頂点間も1手で移動できるようにする①

- 全グループでグラフの中央的な1頂点を共有し、そこから伸ばすようにすると全グループ間が中央を経由して1手で移動できる
- さらにいうと、全グループで1頂点を共有するのではなく、三角形になっている3頂点のどれかを含んでいれば良い
- どうやって実装するかもけして自明ではないが、例えば以下のようにしてやった

中央から伸ばしてどの頂点間も1手で移動できるようにする②

- 「中央の頂点」を根として、距離LBの範囲内でBFS木を構築する
- その木上で、中央の頂点からDFSして、行きがけ順で頂点をグループに加える
- DFSで次に向かう頂点の優先順：子孫に含まれるまだ一度も採用されていない頂点の目的地となる回数の合計が大きいものを優先する
- グループの頂点がLB個になったら中央に戻って再度DFSを始める

- 23%くらいのケースで全部の頂点をカバーできた
- 頂点を全部カバーできなくても、カバーできる範囲だけはこれで埋めて、入らなかった頂点は適当にDFSとかでA上に配置する、でも結構強い

A上で長いパスを作る解法

- LBが小さいと上記の中央集中パターンはうまく動かないので別の方法が必要
- A上の連続した領域がグラフ上で長いパスになっているように構築するとよい
- パスであることで、連続した長さLBの領域どれも連結な頂点集合になって選択肢が増える
- 近くをぐるぐる回るのではなく、遠くまで行っては戻ってくるような経路にすることで、少ない信号操作の回数で広い範囲へ移動できる

旅行パート①

- Aを元に行動を最適化する
- 常に長さLBで信号操作する前提にするとアルゴ問題になる
- 「この頂点から1回の信号操作でここまで行ける」の拡張グラフを作って、その上でBFSする、という感じになる
- ある目的地から次の目的地までの信号操作回数を最小化するだけならBFSすれば良いが、さらにその次の目的地への移動も考えて全体最適にするには結構難しい

旅行パート②

- 自分は妥協して、目的地に到達する最後の1手は、貪欲にその次の目的地までの距離が最小のところに行ける選択肢を取る、としました
- 長さLB以外の信号操作を活用できた人はどれくらいいるでしょうか？

その他実装について①

- 最初に全頂点間の最短距離を出しておくと色々使える
  - O(M)=O(N) なのでワーシャルフロイドよりN回BFSの方が速い
- Aを構築する部分と、構築されたAを使って旅行をする部分とは分離しておくと良い
  - それぞれ独立に改善できる

その他実装について②

- 時間がある限り複数回解いてベストを取るような形だと、答えとしては「連続した移動によってここまで移動する」だけ保持しておけば、具体的な経路は信号操作の情報と合わせて最後に復元でき、毎回復元しなくて良い
  - そもそも問題のルールを、1歩ずつ移動するのではなく、青信号で連結している頂点はどこにでもジャンプできるようにしようかとも思ったが、説明がややこしくなるのでやめた
- ビジュアライザは自作しましょう！
  - 何を表示したいかは解法依存だと思うので