Bus Tour / バスの乗り継ぎ

JOI Spring Camp day1
Shogo Murai

問題の概要

- グリッド状のバス専用道路がある
- 道路の上で、長方形の形をしたコースを延々廻り続けるバスが N 台ある
- 同じ点に来るバス同士は、1 分以上の乗り継ぎ時間があれば乗り継げる
- JOI 君の初期位置と目的地、バスの運行情報が与えられる
- Time to Destination を求めたい

解法へのアプローチ

- 最短時間とか言っているし最短経路問題？
  - Dijkstra 法が使えそう
- 何をキーに Dijkstra をする？
  - （位置、乗っているバス）くらい覚えておけばよさそう
- 位置とバスごとに、そのバスに乗っているための最短時間を求めればよさそう

最短路問題に還元

- (位置, 乗っているバス)を頂点にする
- 辺は、次の2通りに対応するものが考えられる
  - 今のバスにそのまま乗って次の交差点
  - 今の交差点でバスを乗り換える

- いちいちバスの次の交差点を求めるのは大変なので最初にバスの通る点をvectorなどに入れておきましょう
- 交差点ごとにも通るバスをすべてvectorなどに入れておきましょう

計算量の見積もり？

- 位置はO(WH)、バスはO(N)なので頂点はO(WHN)個
- 辺の数は2通りそれぞれについて考える
  - バス移動→頂点数と同じでO(WHN)本
  - バス乗り継ぎ→各頂点から高々O(N)本なのでO(WHN^2)本
- これでDijkstraするとO(WHN^2 log WHN)
  - 30点？

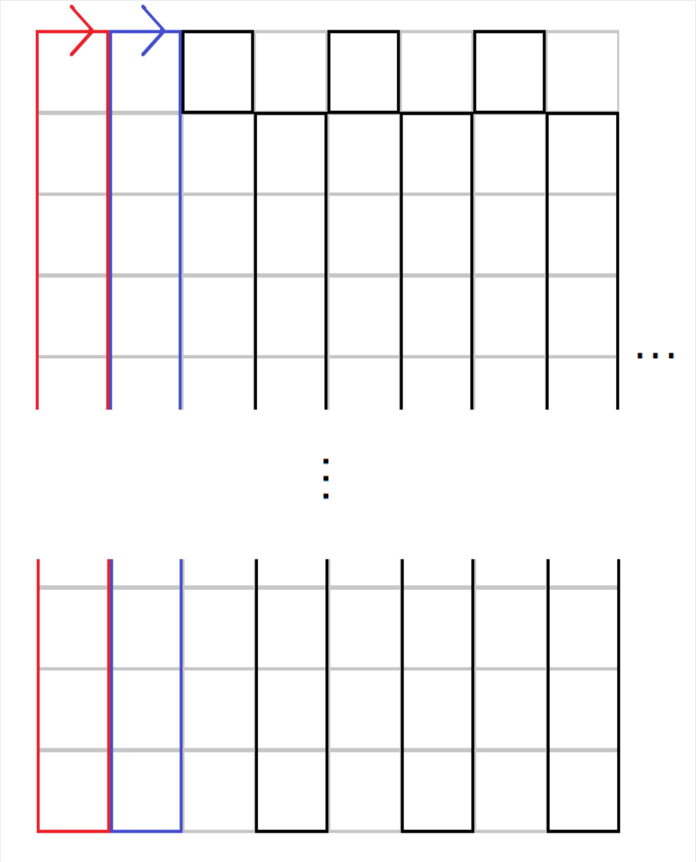
もう少し高速化

- バスはO(W+H) 個の頂点しか通らない
- そのバスが通らない頂点も覚えておくのは無駄
  - 各バスが通る位置だけで考える
- 頂点はO((W+H)N) 個
- 辺の数は
  - バス移動がO((W+H)N)
  - バス乗り継ぎはO((W+H)N^2)
- よってO((W+H)N^2 log((W+H)N))
  - 80点？？

変なテストケースだと
間に合わない

変なテストケース

- 04-04.txt
- 右図のような形をしています
- 赤線と緑線は 400 重になっている
- 最短所要時間も長い
- うまくやらないと TLE する
- こういう TLE 狙いのケースはいつでも存在すると思ったほうがいいです



もっと高速化

- 問題は、考えるべきバス乗り換えが多すぎること
- 同じ点で、2 回以上乗り換え処理は必要？

- 2 回目（遅い方）の乗り換え処理は、1 回目（速い方）の処理をやっておけば十分
- それによってより速い経路が得られることはない
- 乗り換え処理をしたらその点でフラグを立てて、フラグが立っていたら以後は処理をしない

queue も高速化(1)

- 最短時間の最大値ってどれくらい?
- 大雑把に、待ち時間と移動時間の最大値を求める
  - 待ち時間:どのバスでも、3999 分より多く待たされることはないので 3999000 分
  - 移動時間:すべての点を訪れ、全てのバス経路に乗り換える場合が最大で、1000000+999 分
- よって、高々 4999999 分で目的地に到着

queueも高速化(2)

- priority_queueは、所要時間が小さい順に取り出せればなんでもいい
- 所要時間は0～4999999の整数値
- 所要時間ごとに、リストを持って管理できる

- リストは、STLのものよりも自前で連結リストを書いたほうが定数倍で有利です
- このpriority_queueなら、追加はO(1)、取り出しは全部でO(要素数+最大所要時間)

まとめ

- 頂点数は O((W+H)N)
- 辺の数は O((W+H)N)
- Dijkstra が(ここでは) log なしで解ける
- 乗り換えフラグの初期化に O(WH)

- よって、最終的に O((W+H)N+WH)
- これで 100 点
  (log を取らなくても 100 点は得られるようです)

得点分布