AtCoder Heuristic Contest 064

西日本旅客鉄道株式会社
JRW Data Analytics

ALGO ARTIS

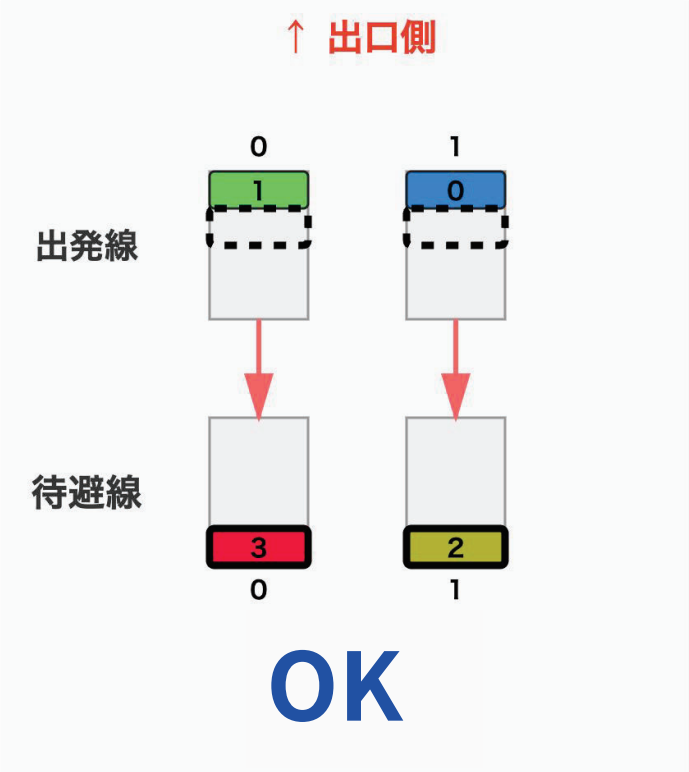
問題解説

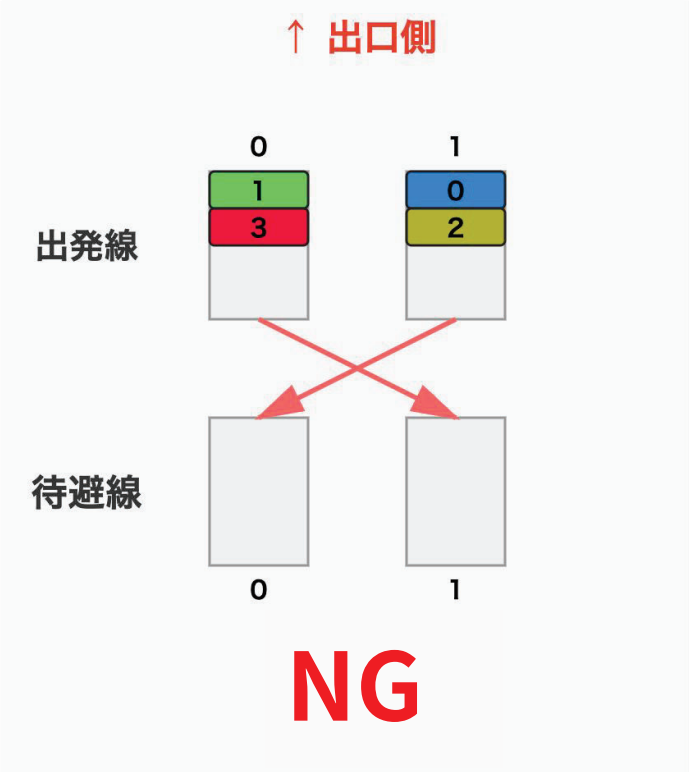
解説放送

2026年4月26日（日）

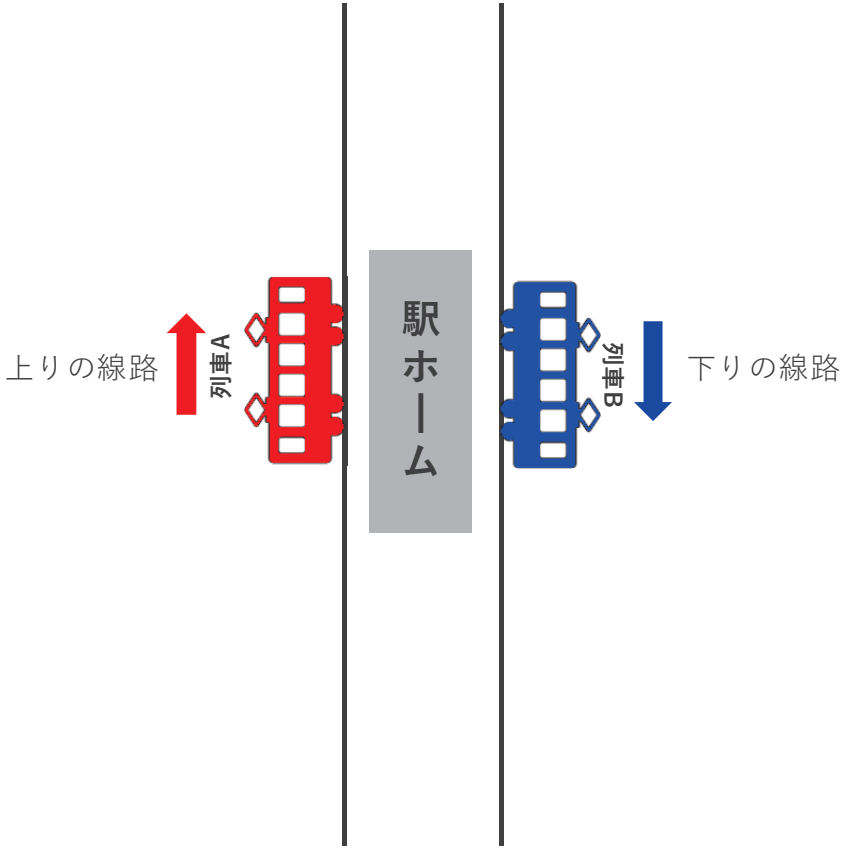
問題のテーマについて

writer陣4人で**鉄道に関連した問題**が作れないか昨年9月頃から検討を開始
最終的に**平面交差**という鉄道運行の制約を題材にした問題を出題することに

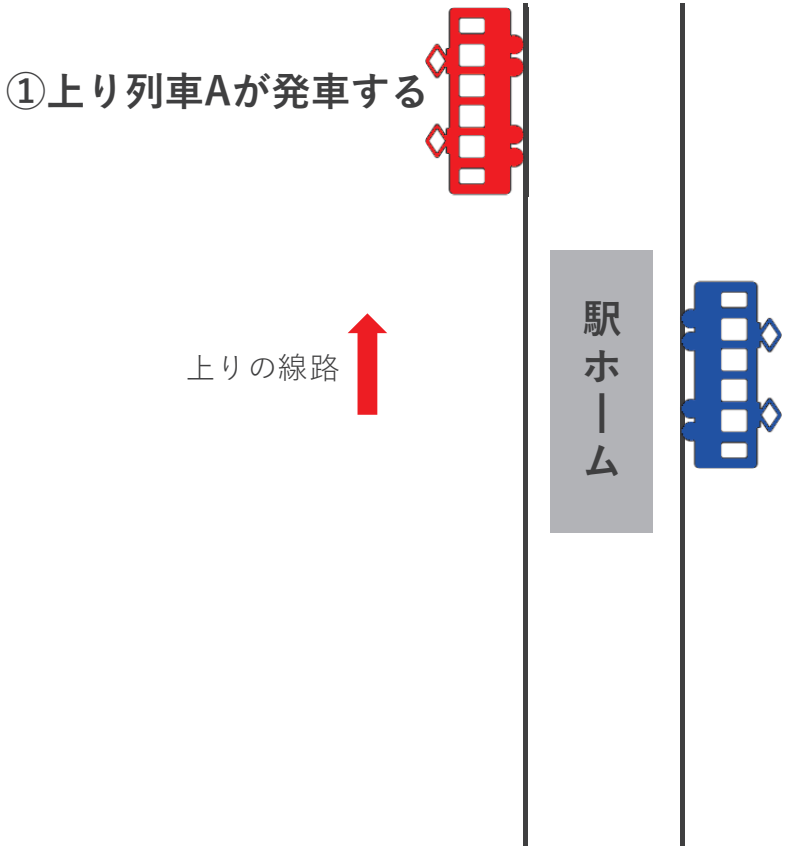




今回の問題は「平面交差」と呼ばれる線路が交差する仕組みがテーマです

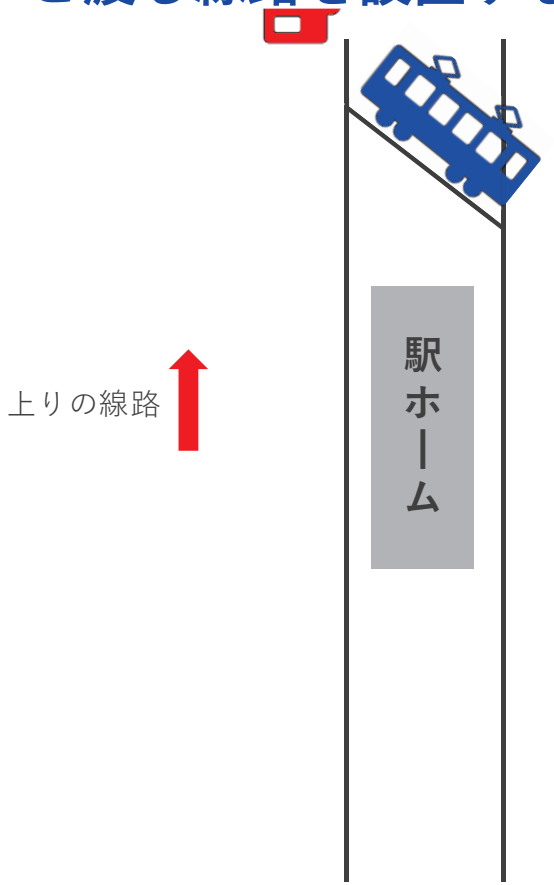


列車には発車順序が決められています



実現したい順序:上りA列車発車→下りB列車が折り返して上り列車として発車

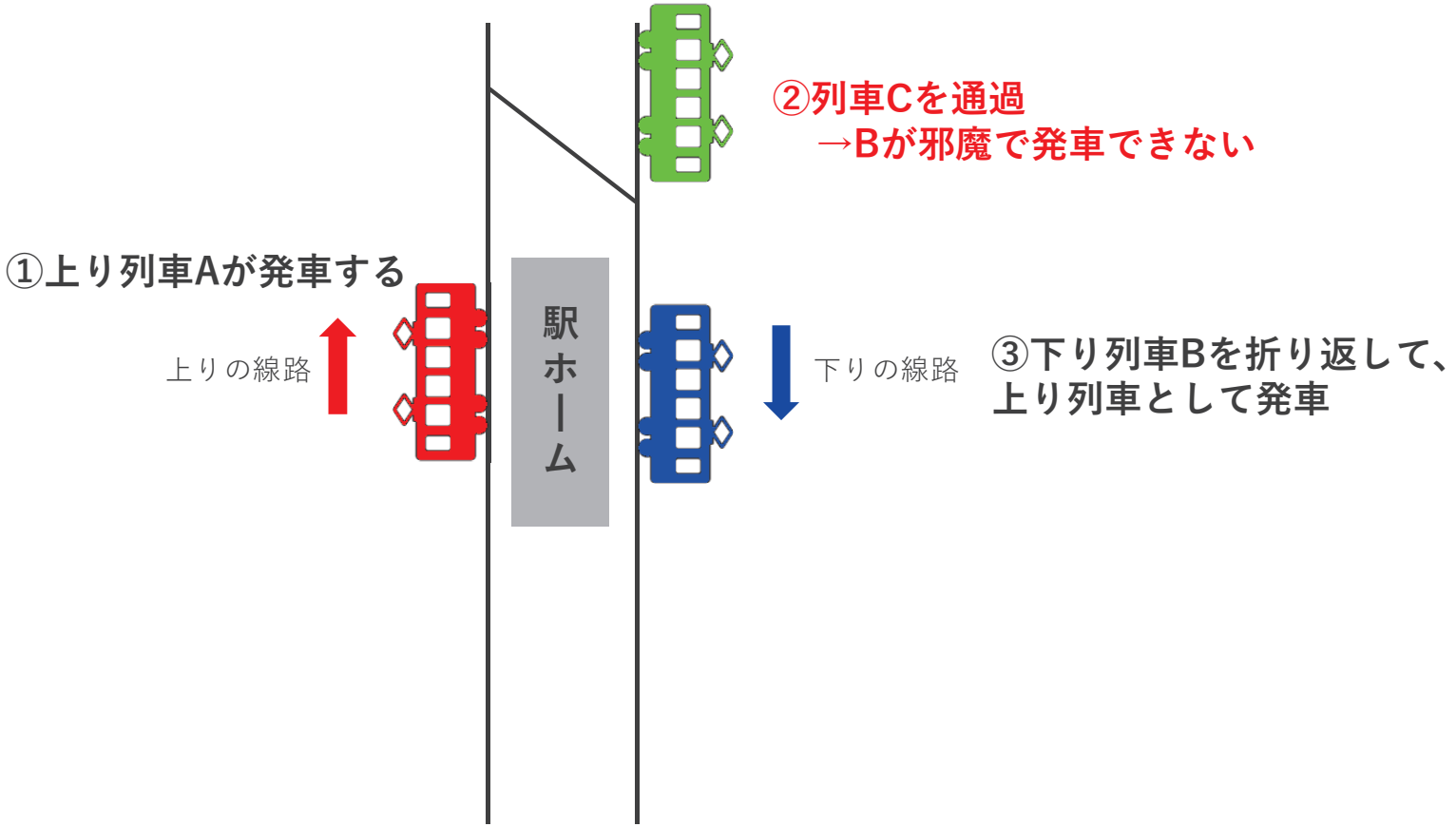
下り線から上り線へと渡る線路を設置することで折り返しが可能になります



ここに分岐器を設置すると折り返し上り列車となれる!

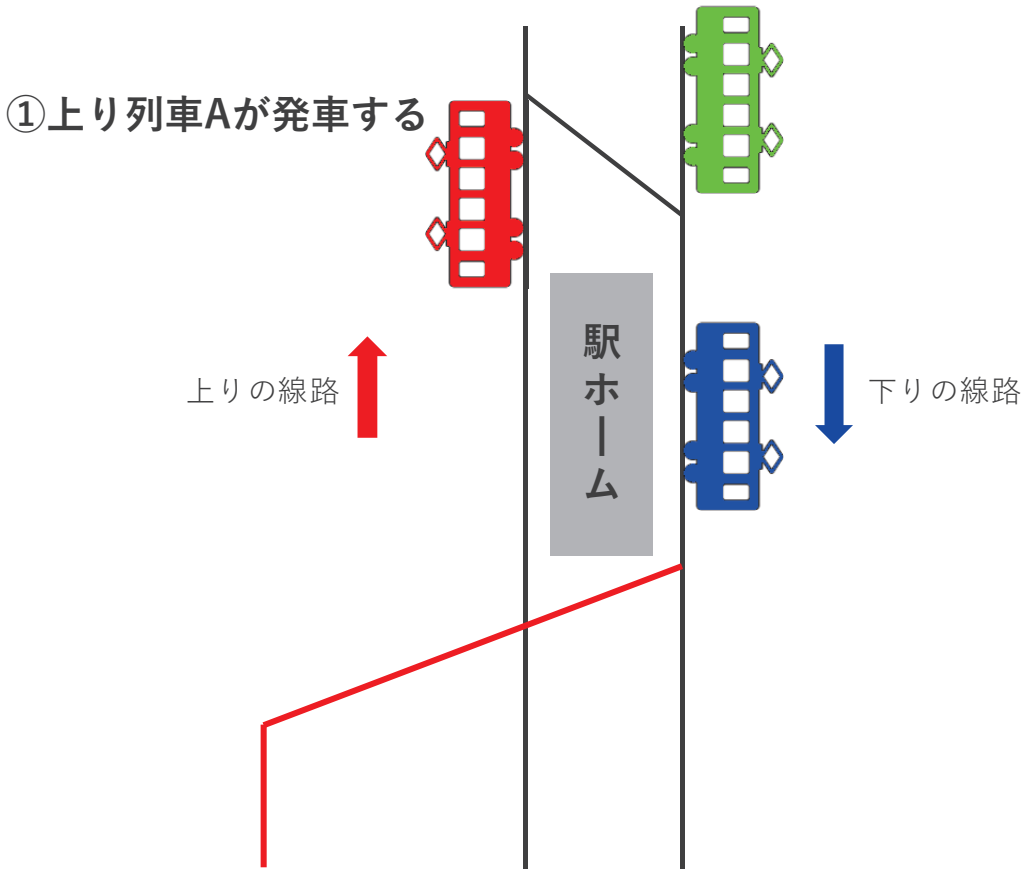
上りA列車発車→下りB列車が折り返して上り列車として発車が実現しました
設備はこれだけで十分でしょうか?

ホームに列車がいるため列車Cが下り線を通過できない

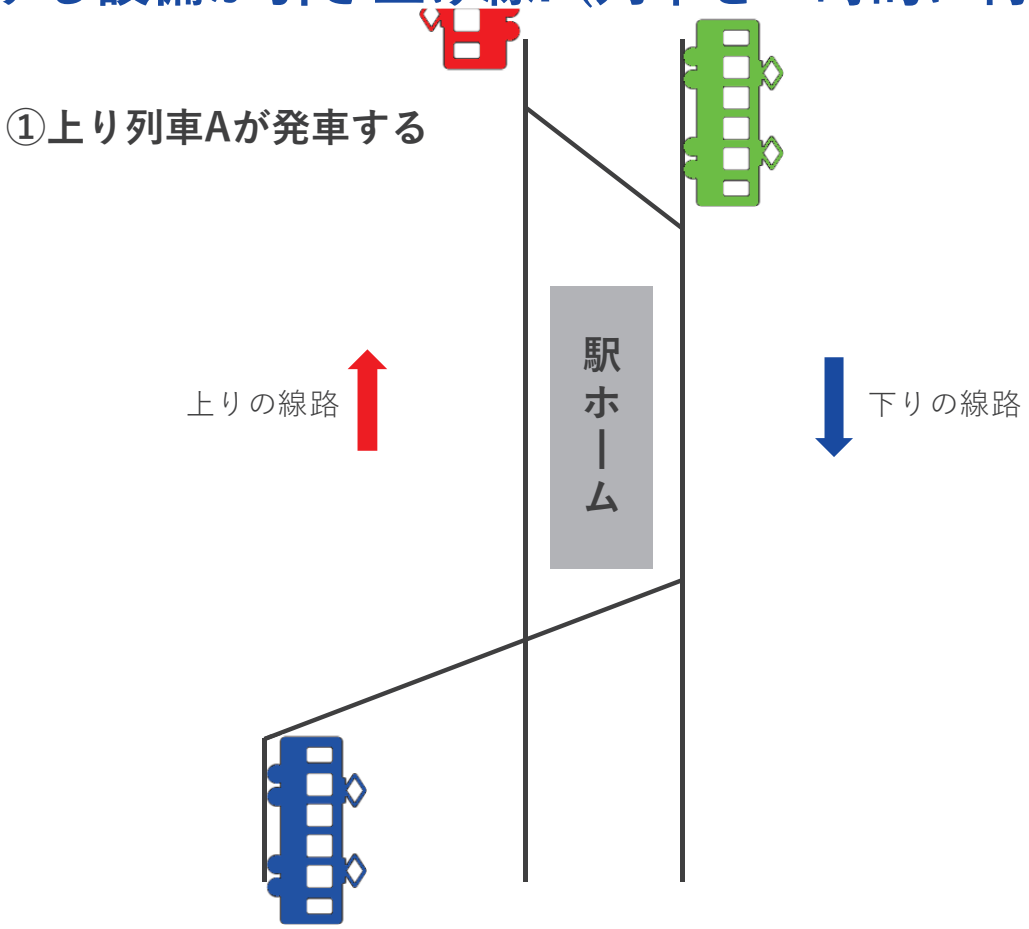


実現したい順序: 上A発車→下C列車通過→下B列車折り返し上列車発車

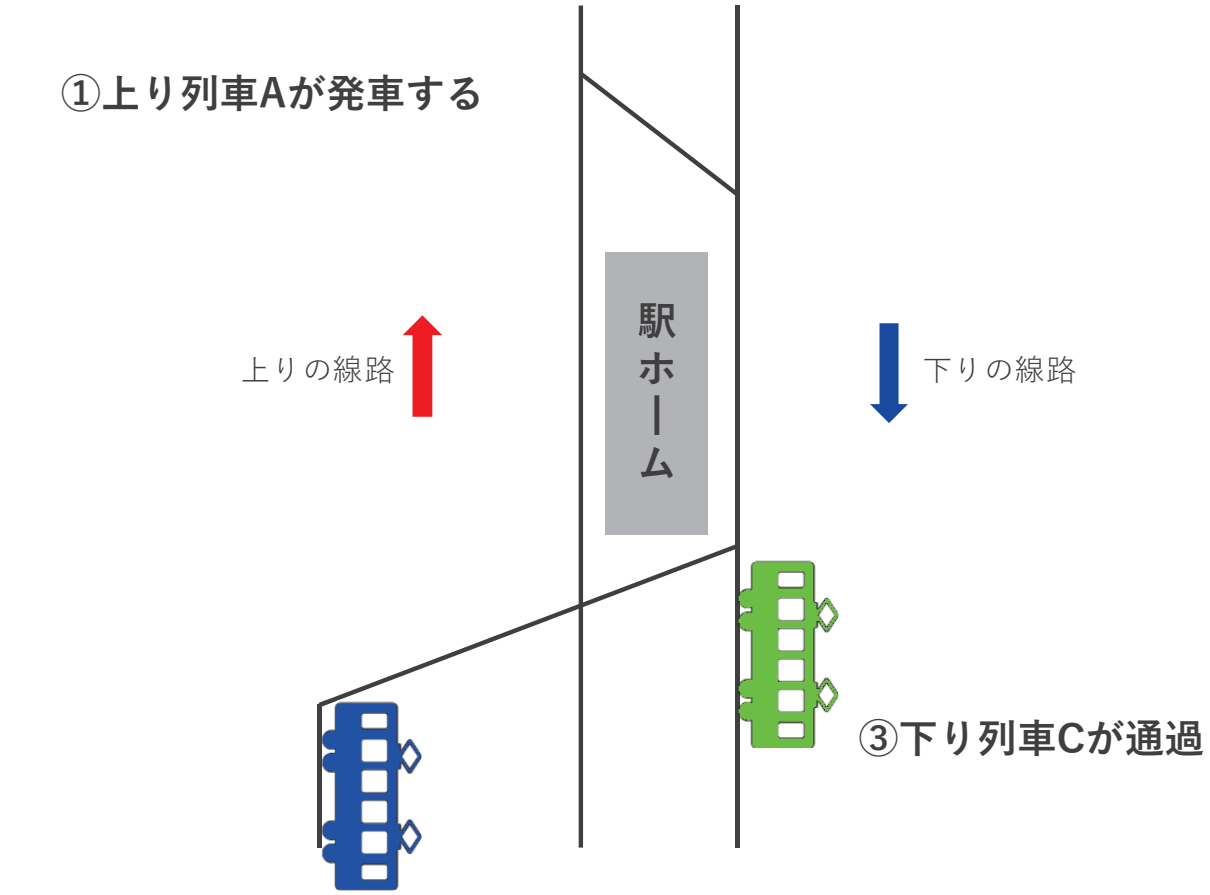
これを実現する設備が引き上げ線!（列車を一時的に待避させる設備）



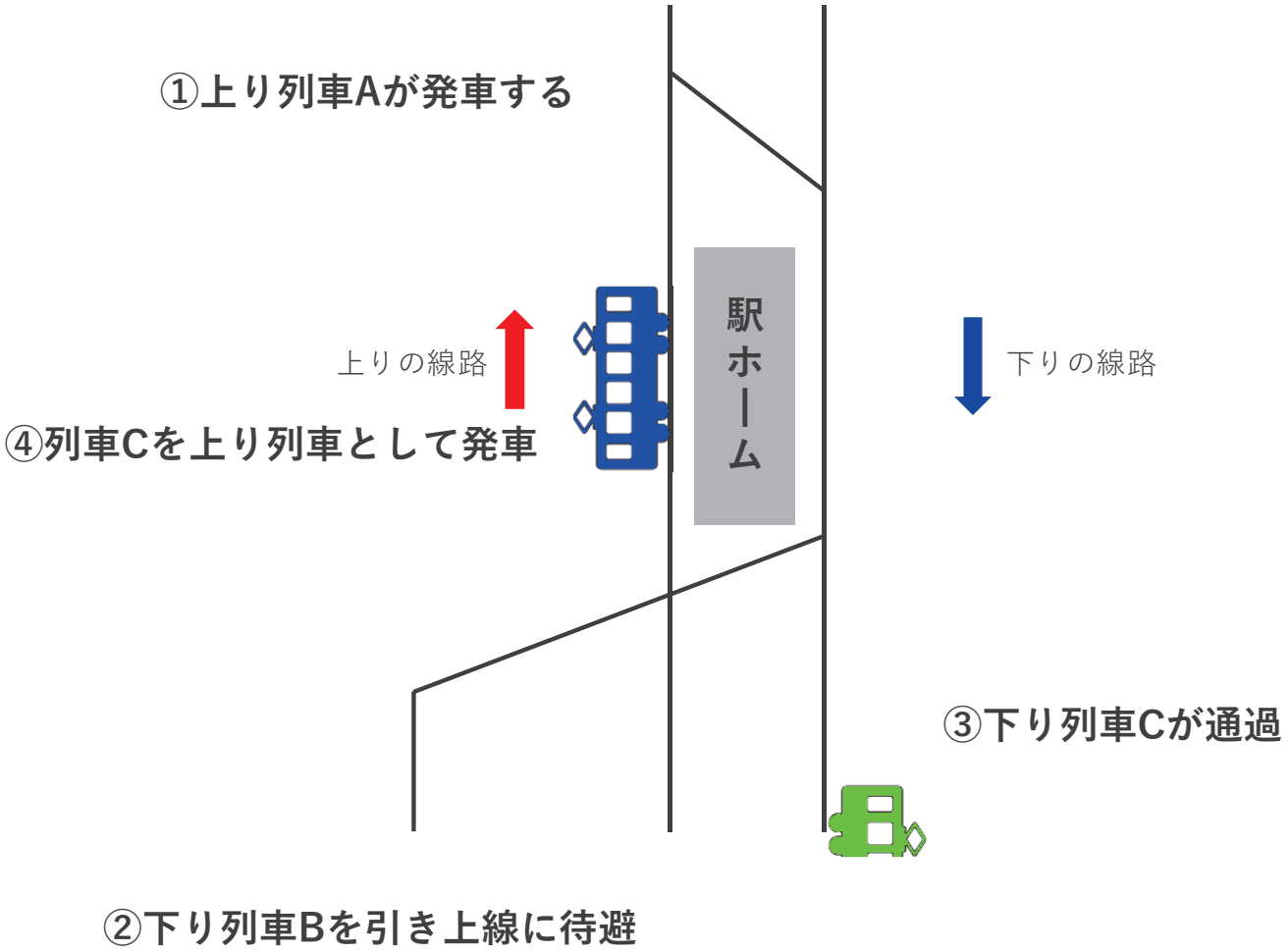
これを実現する設備が引き上げ線!（列車を一時的に待避させる設備）



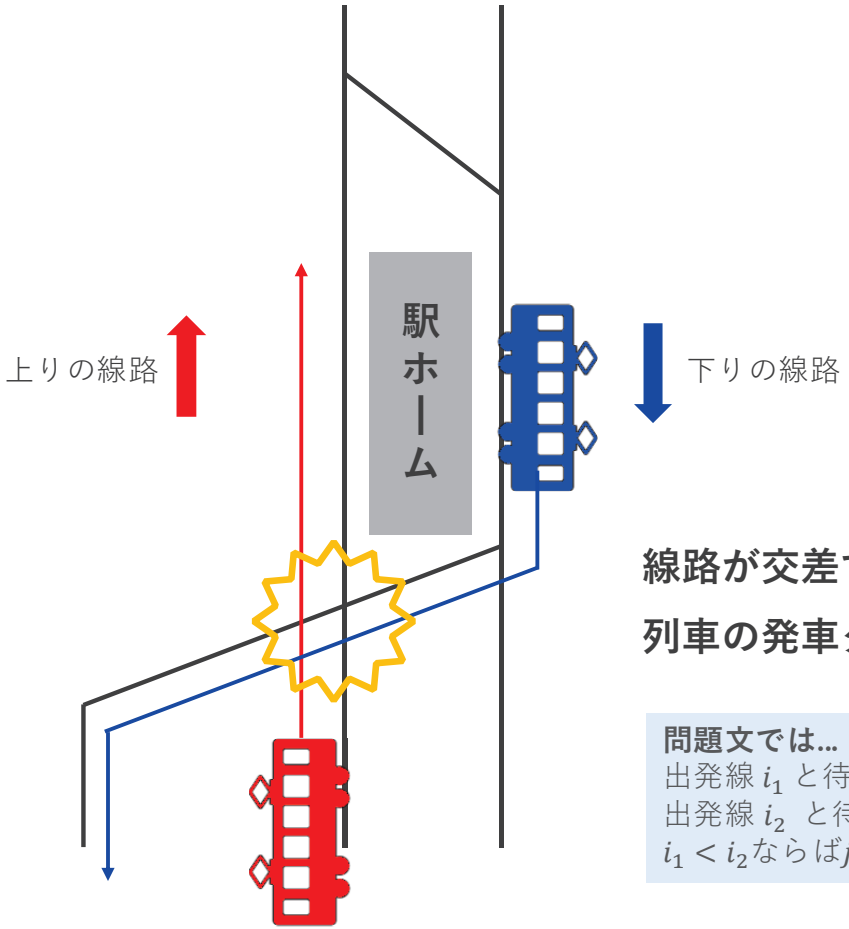
これを実現する設備が引き上げ線!（列車を一時的に待避させる設備）



この設備A発車→C通過→B折り返し発車が実現します



設備を増やすと便利になる反面、制約が発生します



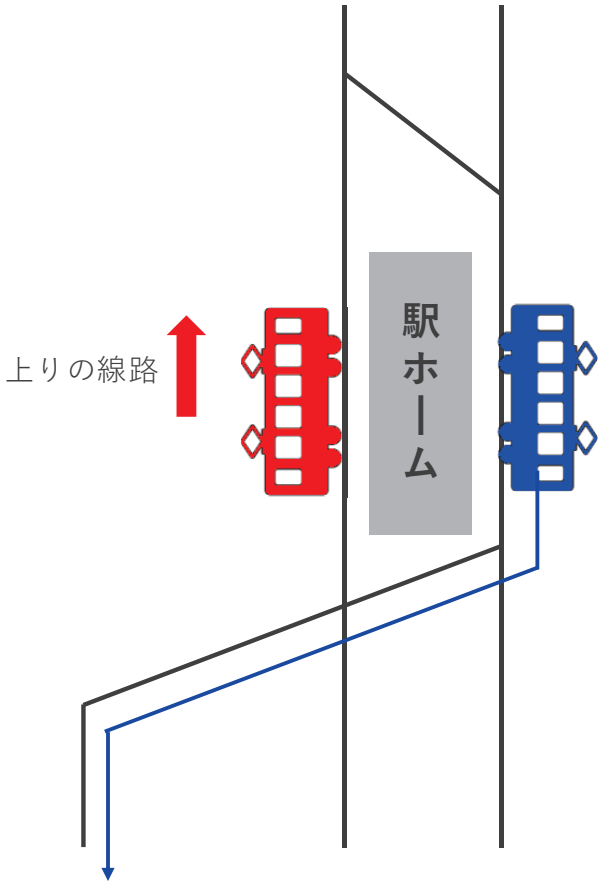
線路が交差する箇所がある(平面交差)

列車の発車タイミング、順序を制御しないと衝突

問題文では…
出発線 i_1 と待避線j_1の間での移動、
出発線 i_2 と待避線j_2の間での移動を同時に行う場合、
$i_1 < i_2$ならば$j_1 < j_2$ でなければならない。

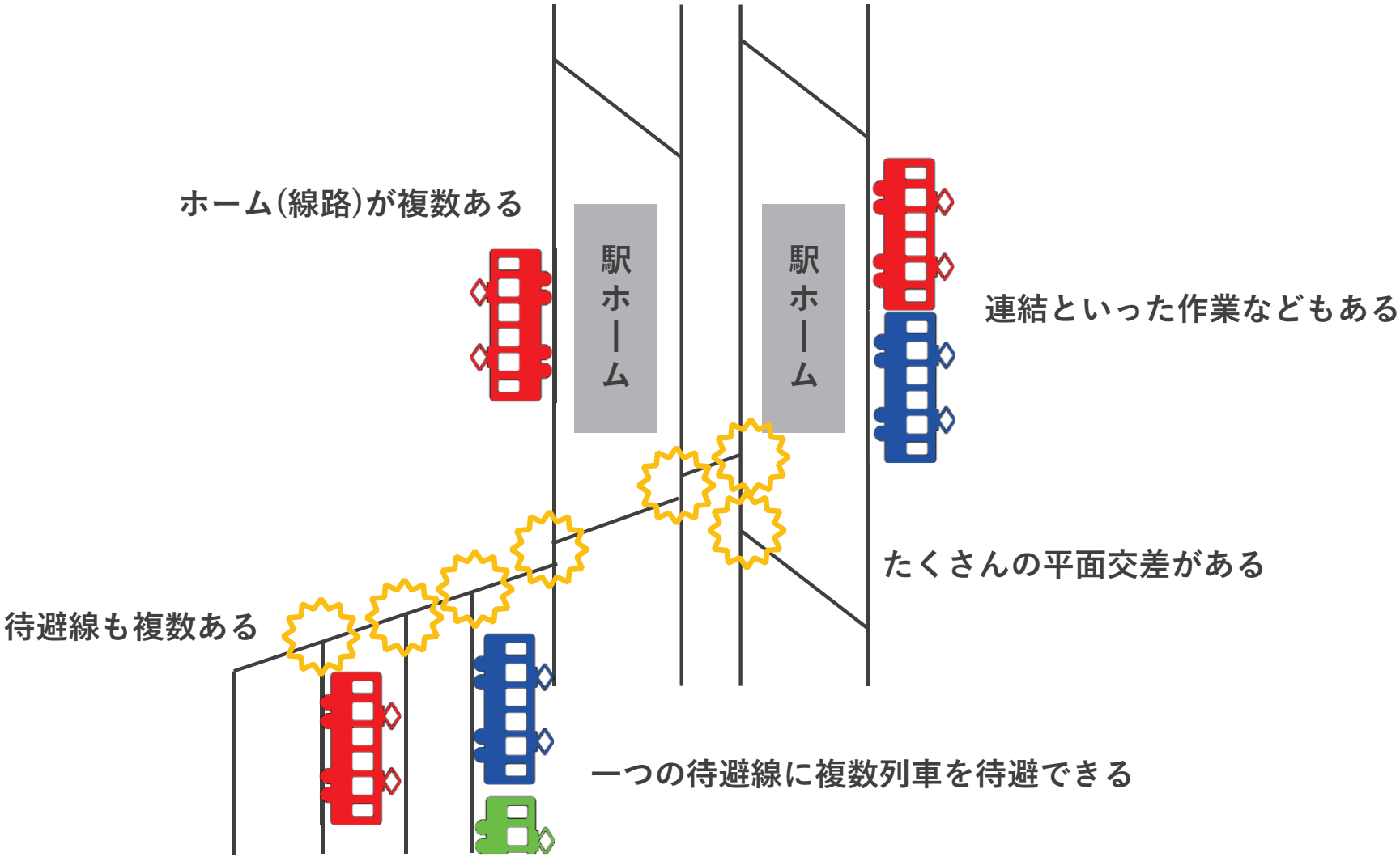
もし引上げのタイミングで後続の上り列車が来ていたら…

実際には信号制御などにより安全を担保しています



上り列車が全両ホームへ停車したのを確認後、
下り列車の引上線への信号を制御

実際の線路はもっと複雑な制約条件があります



平常時はどのような順序で整理するとお客様への影響が少ないかを考えます

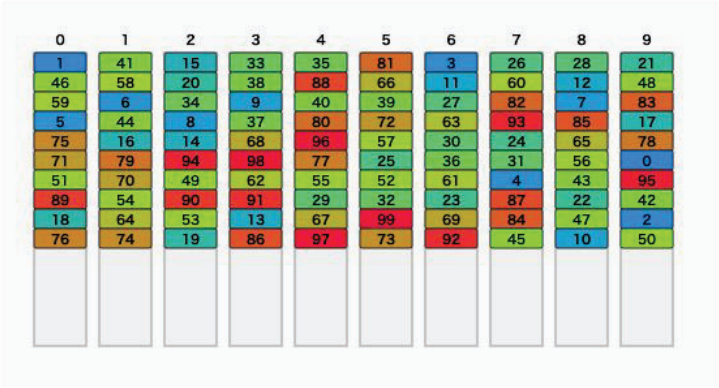
実際の線路
（どこの駅でしょう）

平常時
ダイヤ上で通過順序が決まっています

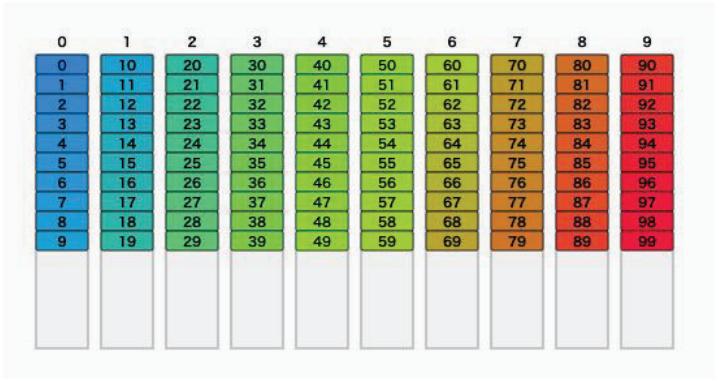
ダイヤ乱れ時
- どの列車を先に通すか、どこで待たせるか
- 進路の競合が発生していないか
- 順序変更によってどのような影響があるか
  - 後続列車
  - 接続
  - 折り返し
  - 車両運用…
  - 乗務員の運用…

鉄道の「最適化」の難しさとできたときの社会へのインパクトが少しでも伝わると嬉しいです！

最初の考察



並び替え



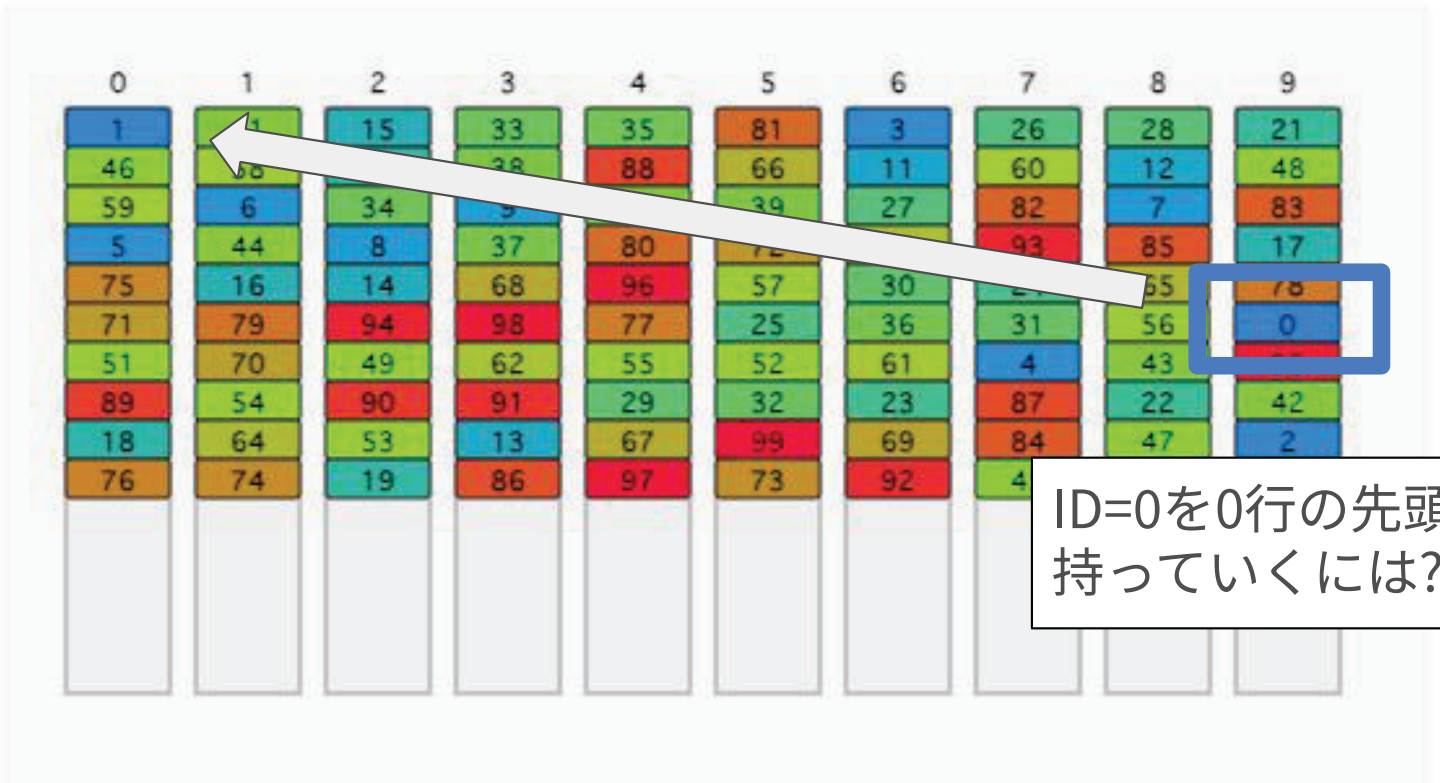
- 2023年11月の**AHC026**のような問題設定
  - 複数の箱のスタックが並んでいて、一番上から箱を取り出せる
  - IDが小さい順に200個の箱を取り出す
  - 上位解では**箱をソート**してから一気に取り出す方針が有力

- 筋の良いルールを見つけることができないか?
  - **1列ずつ順番**に作ってみる?
  - 待避線側に**ソート**してみる?

まずは簡単な方針を実装して観察をする

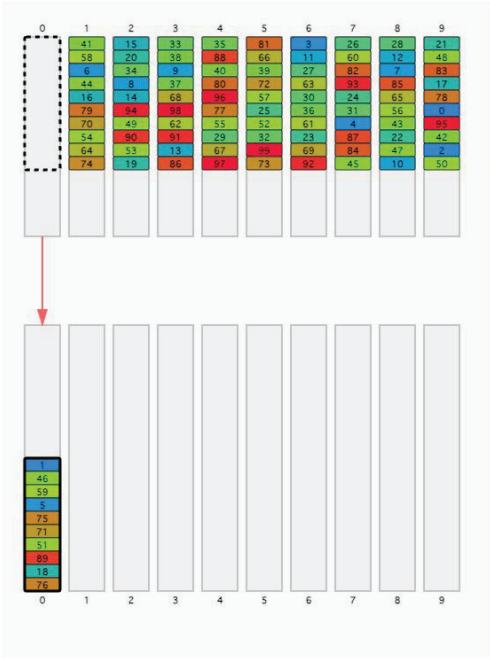
解法(1) 1両ずつ順番に並べていく

ID 0の車両から順番に1両ずつ並べていく

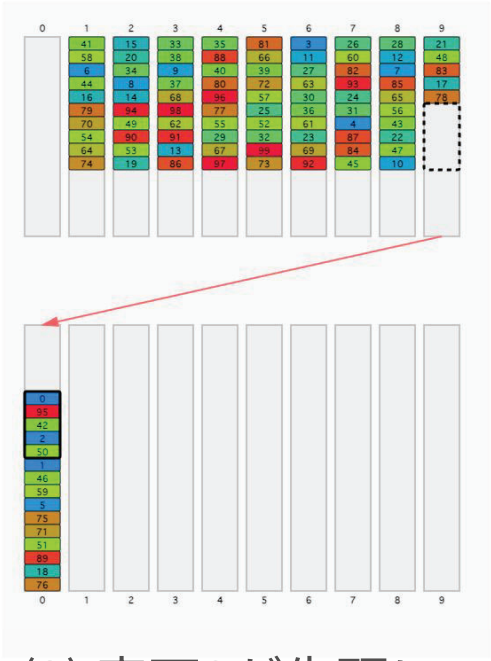


解法(1) 1両ずつ順番に並べていく

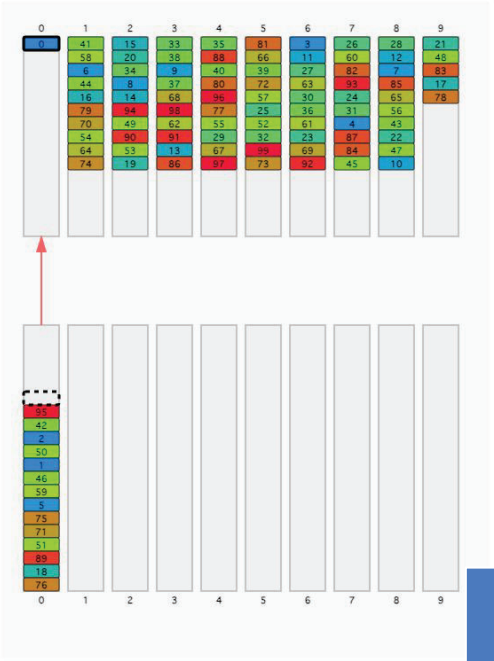
目的の車両が待避線側の先頭に来るように車両を移動させる。 1両を正しい位置に持っていくのにおよそ**2手**かかるので、およそ**200ターン**で並び替えが完了



(1) 列0を空にする



(2) 車両0が先頭になるように移動

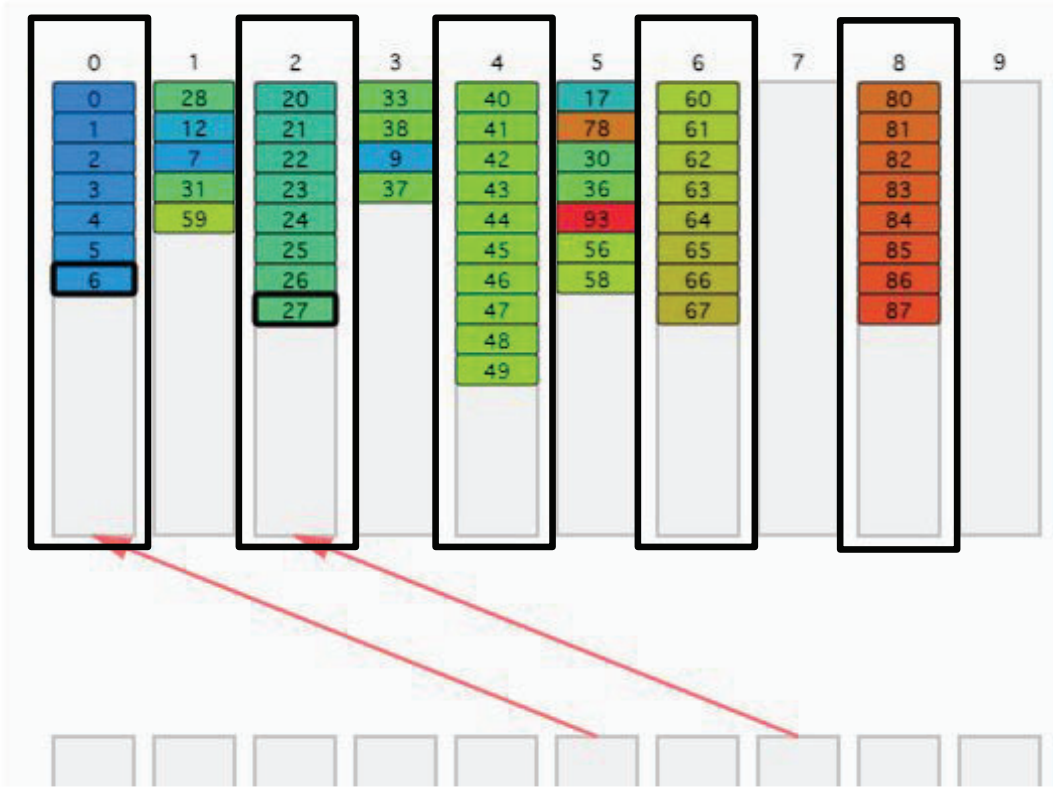


(3) 車両0を移動

719,563点
(平均202.9手)
本番593位相当

解法(2) 解法1を並列に実行

1ターンに複数の車両を動かすことができるので、**複数の列を対象**にして解法1を実行。ターン内で可能な限り多くの車両を移動させる。

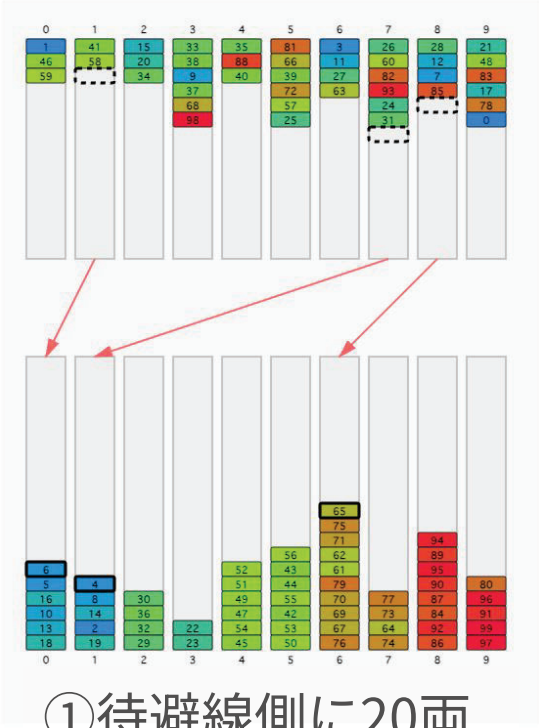


- 最初に偶数列を対象にして、順番に移動させる
- ターン内ではできるだけ多く移動をする
- 偶数列が完成したら奇数列を作る
- **交差の制約が厳しい**

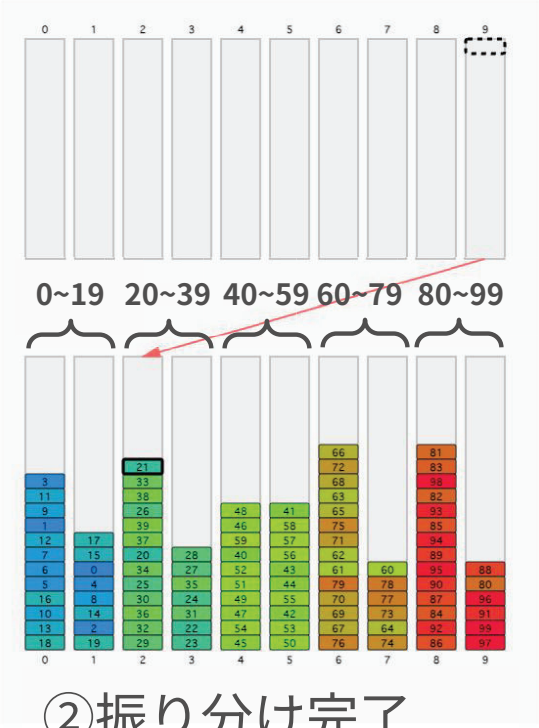
733,733点
(平均108.4手)
本番379位相当

解法(3) 待避線側に2列ごとに割り振りをする

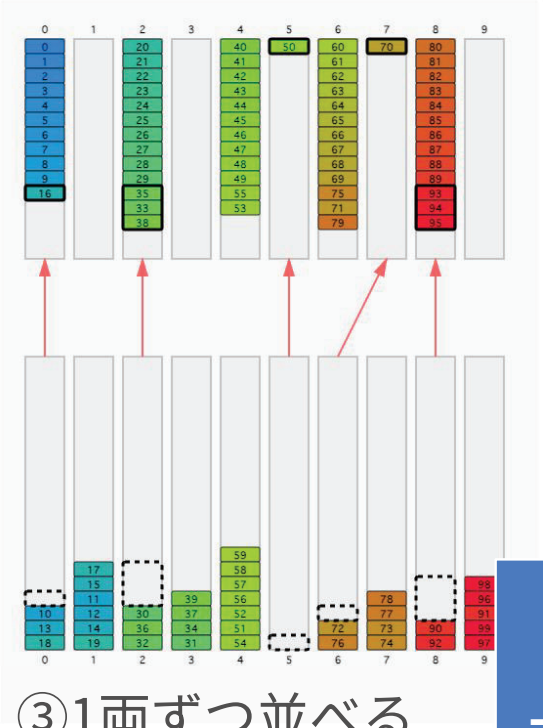
各車両を ID の範囲ごとに **2 列単位の待避線へ振り分ける**(例: ID0~19は列0・1、ID20~39は列2・3)。その後**2列ごとに解法1の方法で並び替える。**



①待避線側に20両ずつ振り分け



②振り分け完了
およそ25手かかる



③1両ずつ並べる
およそ40手かかる

740,147点
(平均65.7手)
本番133位相当

考察

並び替えの解法だと限界が見えてくる。方針を大きく変えなければいけない

- 解法2の方針では、うまくやっても1ターン内での平均の移動数が3を超えることことは難しそう。**有効な移動を200回**実行しなければいけないことを考えると、60~80手くらいが限界に思える(と思っていました)

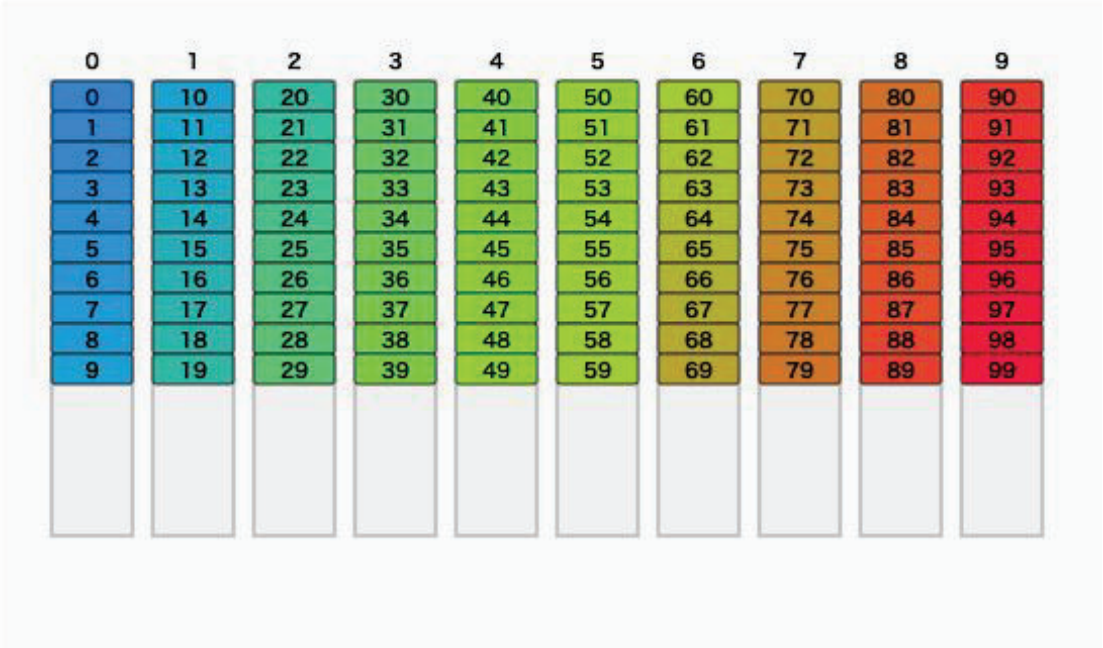
- 1列ずつ完成させていく方法では、**交差の制約が厳しくて**効率的に移動できていない
  - 必ず車両を正解のところに移動させないといけない、というルールを課したことで非効率な移動を強いられている
  - ID=0の車両が9列目にいると、他の車両の動きを完全にブロックしてしまう

- 解法3はぱっと見悪くないが、伸び代は少ないように思える(と思っていました)

※ 元ネタを出したyunixはここで詰まって相談したら、Jirotechが次ページの考察を出してくれました

考察

盤面の良し悪しを評価関数にして、評価値を改善する行動をとることを考える



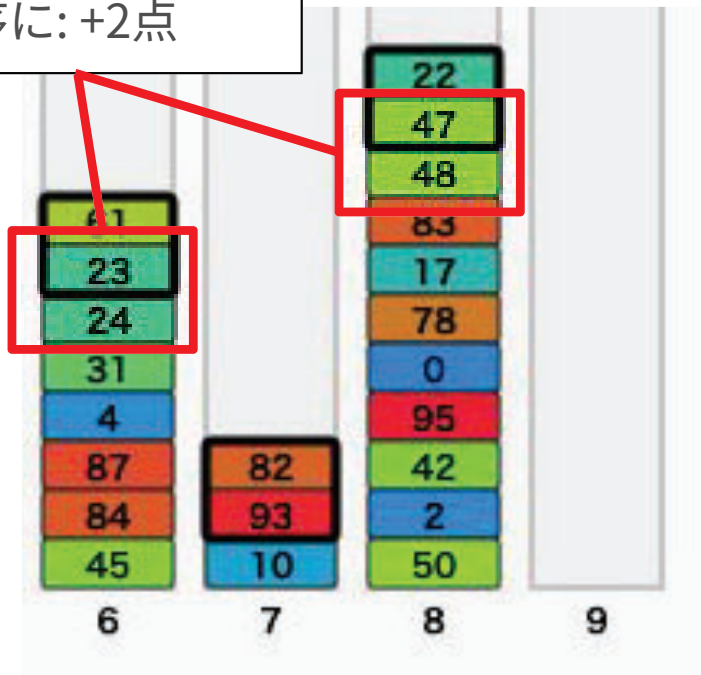
最終盤面では
各列が順番通りになっている

↓

100両の各車両について、**一つ前に正しい車両が配置**されている、または**先頭**にいる

考察

車両を並び替える問題ではなく、**100両の車両を順番にくっつける**問題と読み替えて、評価関数を良くする行動をとる



評価関数の基本的な考え方

- 23-24のような正しい並びに+1点
- 出発線側i列目の先頭に10iがいる時に+1点
- 合計100点になると完成

1ターンでの行動の決め方

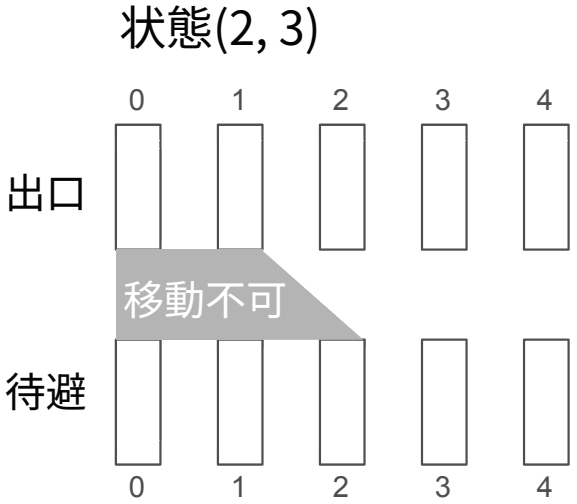
- できるだけ多くの得点が取れるように動かす
- 交差しないように複数の車両を動かす方法は色々
- ここではDP解法について解説

評価関数と**1ターン内での行動の決め方**の二軸での工夫が必要

解法4. ターン内の移動を DPで決定

ターン内の評価関数を最大にする**移動の集合はDPで決める**ことができる

- 毎ターンごとに**左の方から移動で使用する線路を決めていくようなDP**によって移動の集合を決める
- DPはR×Rだけ状態数がある(Rは列の数)
  - 状態(i, j): 出口線のi-1列目、退避線のj-1列目までの線路を移動で使用している状態を同一視して、最も高得点なものを残す
  - 出口線のi列目・待避線のj列目**以降**を利用して車両を移動可能

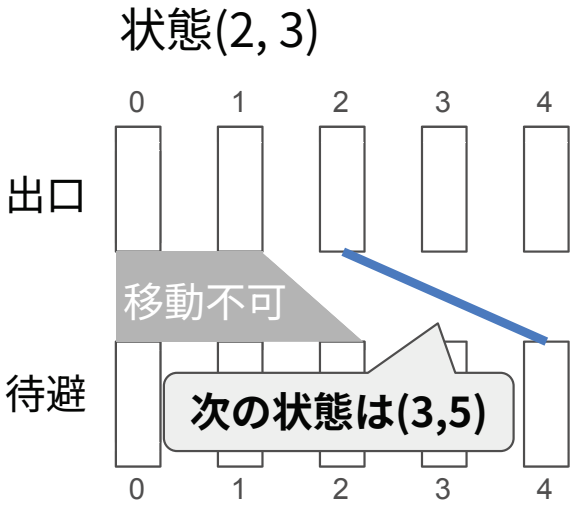


解法4. ターン内の移動を DPで決定

ターン内の評価関数を最大にする移動の集合はDPで決めることができる

- 次の状態への遷移は**選択された移動によって決定**される
  - 出口線i列目・待避線j列目以降を使用する全ての移動を試す
  - 出口線i2列目・待避線j2列目を使用する移動が選択された場合には状態(i2+1, j2+1)に遷移

- 評価関数は正しい車両の並びに対して+1点、100点になったら完了する最もシンプルなものを採用
  - DPの状態の中で大量に同点が発生するので、乱数でタイブレーク
  - 実はこれだけだと全然ダメ...

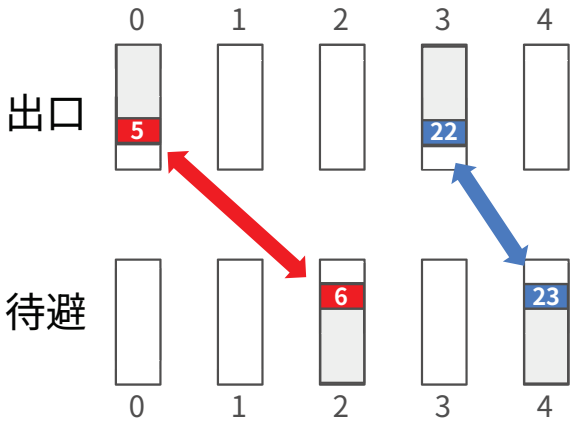


725,816点
(平均161.2手)
本番544位相当

解法5. DPの評価関数を工夫する

次の1手で連続する車両をくっつけることができる状態に加点をすると改善

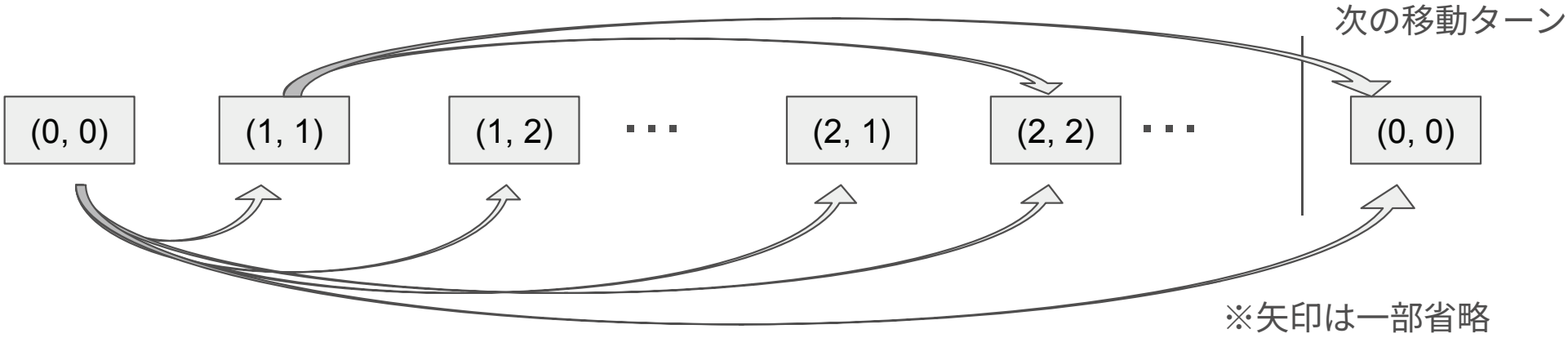
- DPは1ターンの中での得点(=連続した車両をくっつける数)を最大化できるが、**そもそもそのような状態にある車両配置が滅多に存在しない**
  - ちょうど運よく連続する車両ペアが上下に分かれて存在していないといけない
  - タイブレークのランダム要素に祈るしかない
- **次の1手で連続になる配置に加点する**
  - 上下でIDが連続する車両ペアに**0.5点**を加点する
  - 時間いっぱい回して、平均40手程度と大幅に改善する



743,782点
(平均41.5手)
本番9位相当

解法6. ビームサーチ化する

DPの状態ごとに同じ得点でタイブレークが発生していた。**DPの状態をターンとするビームサーチ**にして複数の候補を保持できるようにする。

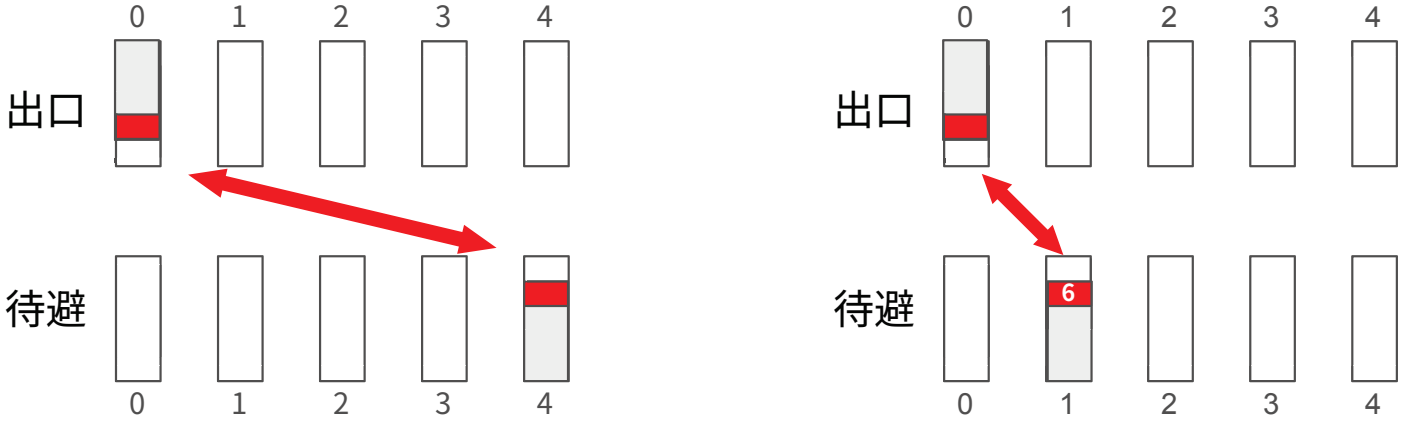


- DPの状態をビームサーチのターンと見立てて、「**ターンが飛ぶビームサーチ**」をする
- 差分更新で実装すると幅200くらいになった
  - 各移動ターン(問題文のターン)では必ず(0,0)を通るので、(0,0)の状態だけ幅を10倍する工夫を入れた

745,784点
(平均28.1手)
本番1位相当

解法7. 評価関数の工夫

「次の一手でくっつくペア」に対して**距離に依存する評価**を追加・**目的地との距離**を車両ごとにペナルティで追加



- 前頁までの評価関数だと、図の左右の状態を区別せず一律に0.5点を加点していた。しかし、**右の方が他の列の移動を邪魔しないので良い状態**のはずである。「次の一手でくっつくペア」に距離に応じてペナルティを設定した。
- 最終目的地の列との距離に応じたペナルティを設定

746,076点
(平均26.2手)
本番1位相当

おまけ: yochanによる焼きなましベースの解法

ホワイトボードでyochanが解説します!

745,858点
(平均27.6手)
本番1位相当