

SRM573 div1 medium解説(自分用)

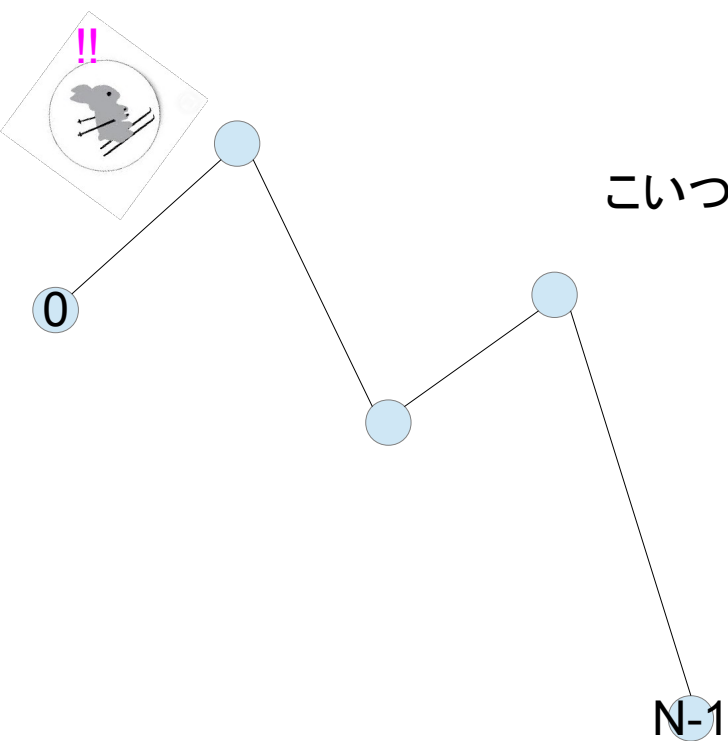
- ・ 問題

- ・ スキー場がいくつかあって、それぞれのスキー場がつながってたりつながってなかったりする(用はグラフが与えられる)
- ・ 0番目のノードからN-1番目のノードに行きたい
- ・ ノード間を移るとき、高いところからそれ以下の低いところに行くようにしか移れない
- ・ ノードの高さを(あらかじめ)変えてもいいがコストがかかる
- ・ 0番目からN-1番目までいけるなら最小コストを、いけなければ-1を返せ

考察

- つながっているかどうか知りたい
- 複数の経路が存在した場合、それぞれの経路について最小コストを考えなければいけない
- 最小コストを求める方法を考える

最小コスト

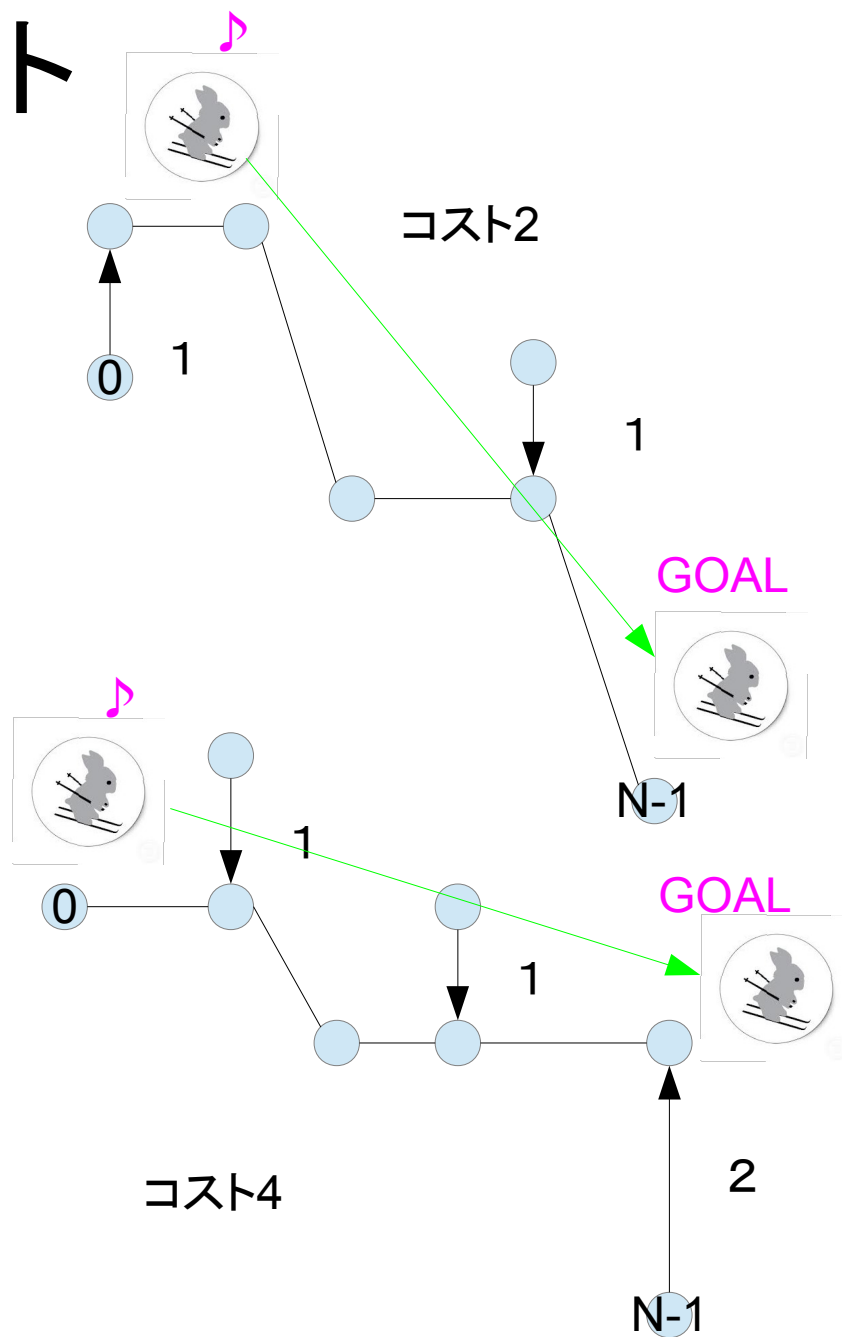


こいつを

こうとか

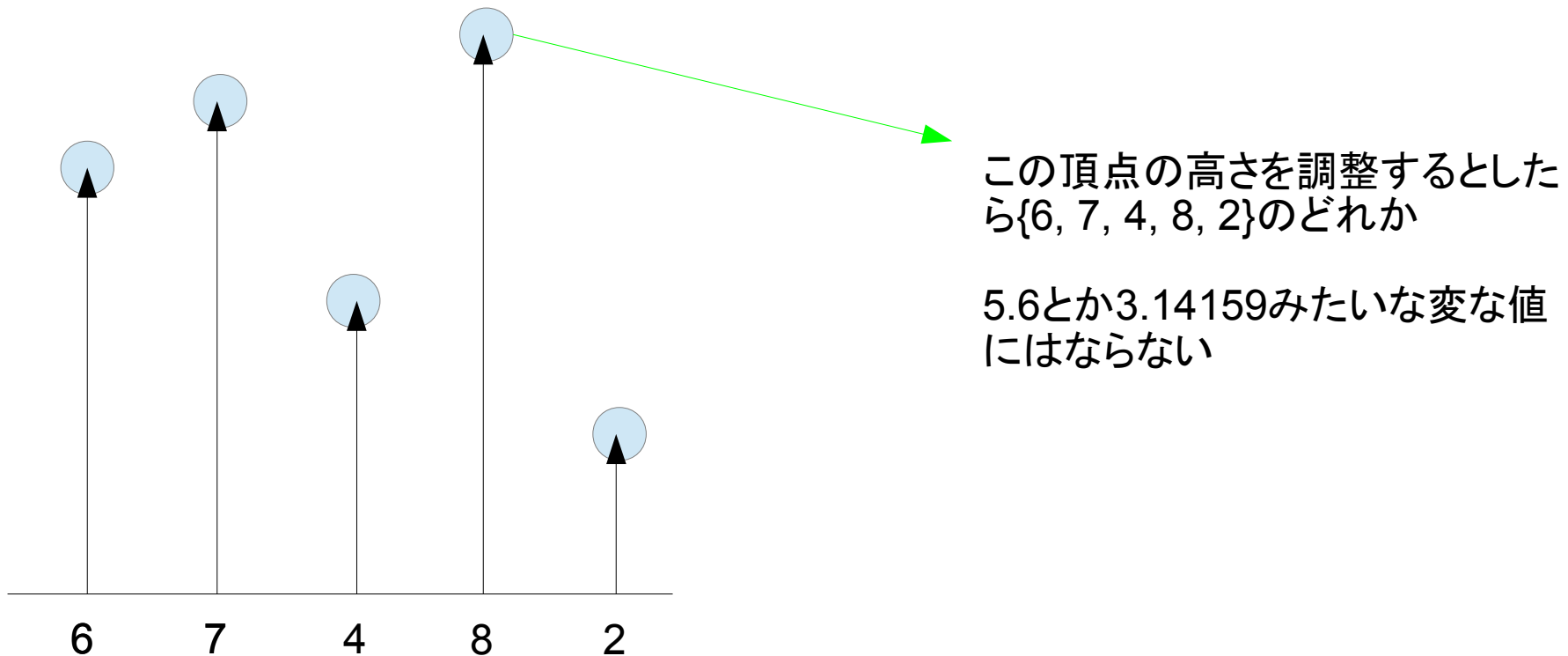
こうとか

それぞれの頂点の高さを調整して、高さが単調減少になるようにする。



頂点の高さの調整

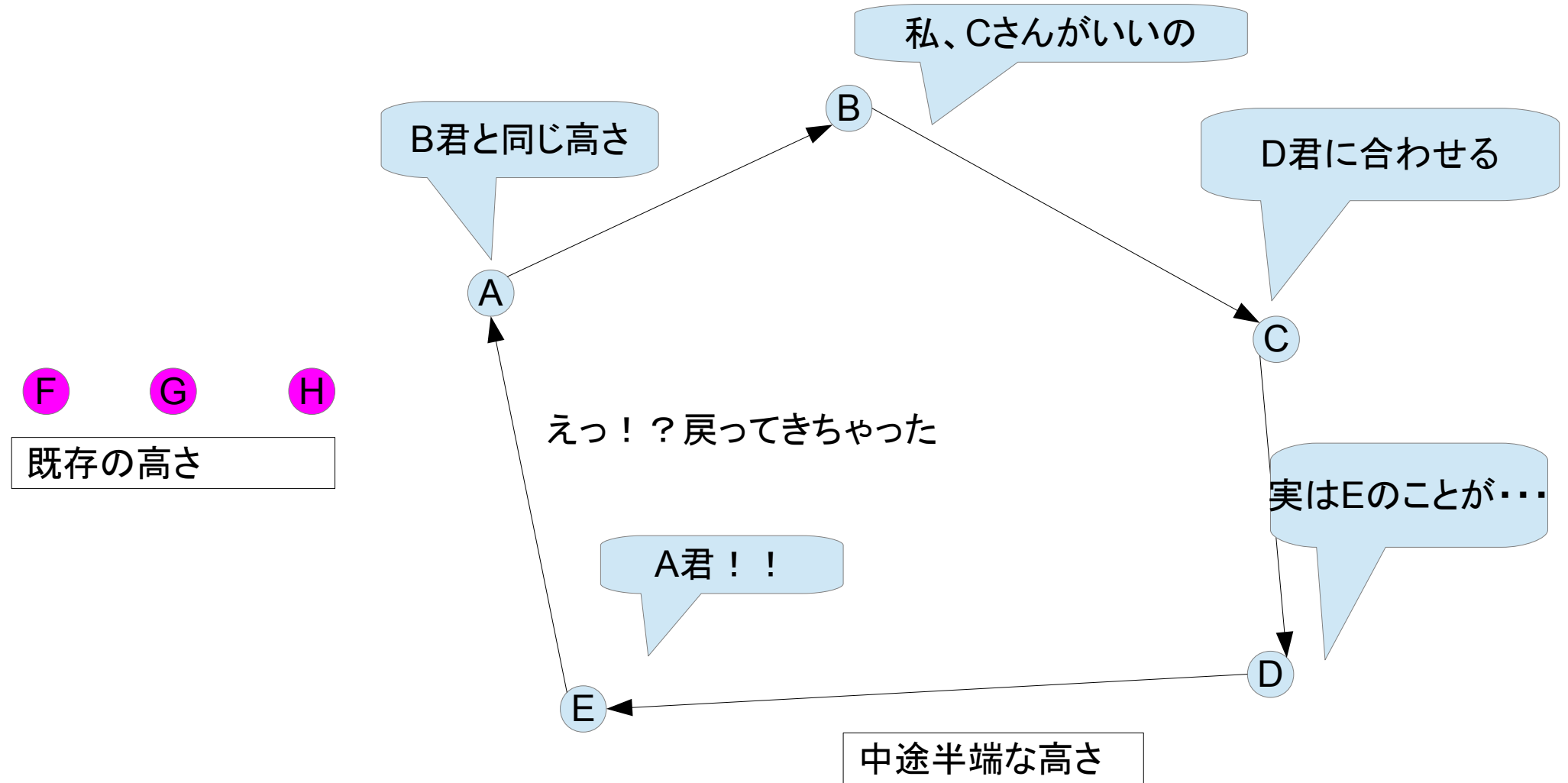
- 調整後の頂点の高さの候補は、最初の頂点の高さになる(たぶん自明ではない)



最初の頂点の高さを候補にすれば十分な理由

- ある経路について、頂点の高さを適切に調整して、単調減少の条件を満たしたと仮定する。
- 最小コストになるように調整したとき、すべての頂点の高さが既存の高さのどれかになっていることを示す。
- 調整前の頂点の高さの集合の要素を”既存の高さ”とし、既存の高さでない高さを”中途半端な高さ”とする
- 条件を満たしている調整後の経路について、中途半端な高さをした頂点がある場合を考える
- 最小コストになるように再調整していく
- この頂点を元の高さへと再調整すればコストは低くなるので、限界まで元の高さへ再調整する。
 - もし元の高さへ戻しても条件を満たしているのなら、OK(既存の高さになった)
 - もしもとの高さに戻らないとすると、限界までコストを下げたときの高さは、他の頂点と同じ高さ
 - 高さをあわせた他の頂点の高さが既存の高さなら、OK
 - 他の頂点が中途半端な高さなら、その頂点も高さの再調整を行う
- 高さの再調整によって、ある頂点の高さを次のいずれかにあわせることができる。
 - 元の高さ
 - 他の頂点の高さ

再調整を繰り返していくと...



中途半端な高さの頂点は閉路を作る

閉路の高さをまとめて再調整

- 閉路に含まれる頂点の相対的な高さを変えずに全体の高さを一斉に変える。
- それぞれの頂点について、元の高さより高いものの個数を a 、低いものの個数を b とする(同じ高さでないことは前述から保証されている)
- 全体の高さを微小量、上げ下げすることを考える
 - $a < b$ なら全体の高さを上げた方がコストが下がる
 - $a > b$ なら全体の高さを下げた方がコストが下がる
 - $a == b$ なら上げても下げてもコストは下がらない、じゃあ動かしてもいいや
- 上げ下げしていくとどっかで、コストが極小になる
 - どこかの頂点が既存の頂点になったとき
- 結局、以下のいずれかが常に成り立つ
 - コストを最小にするためにすべての頂点を既存の高さに揃えなければならない
 - すでにコストは最小だが、コストを維持したまますべての頂点を既存の高さに揃えることができる

再び問題へ

- 要は、それぞれの頂点を何処かの高さに揃えながら経路を探していく
 - 直前の頂点の高さを超えないように経路をつなげていく
 - 頂点の高さを再調整したときはその分だけコストを加算
 - つまり、頂点 $node(N)$ と高さ $height(N)$ のpairをqueueに突っ込んでいく
 - costの小さいものから調べて行って最初に見つけたものが最小コスト
- `priority_queue<pair<cost, pair<node, height> > >`
 - 大小関係を定義するのが面倒くさいので、costを負にしてqueueに入れておくと最小コストが最大になりtopで取り出せる
- 結果
 - 195ms
 - 短く書いたらけっこう時間かった
 - 450ptだけど500ptでもいいような...