

# 第11回Asprovaプログラミングコンテスト (AHC037)

# 元ネタ

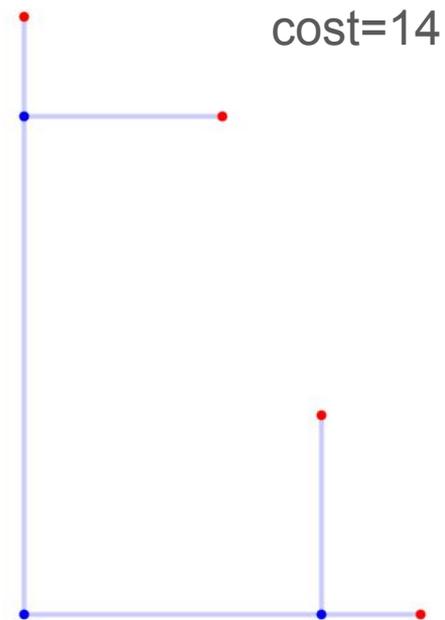
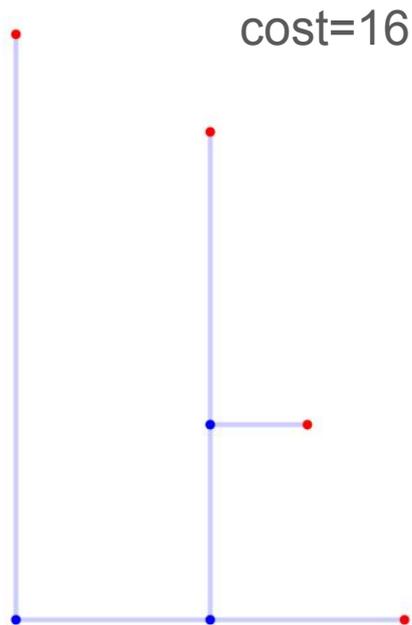
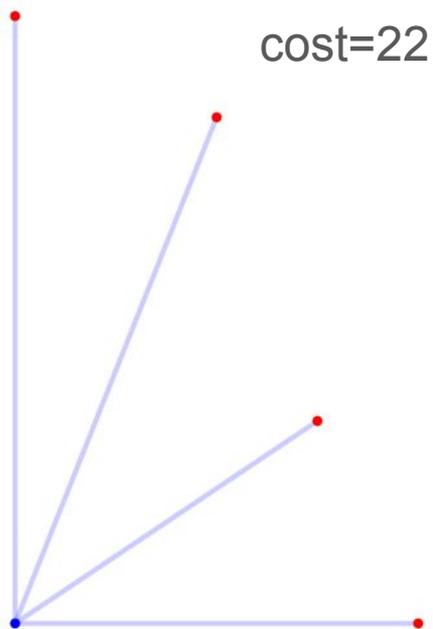
こんな感じの問題を作る方向で考えた

- 1つの原料を分割していろいろな製品を作ります
- 複数の製品に共通する処理は分割する前にやった方が効率的です
- なるべく分割のタイミングが遅くなるように計画してください

ストーリーは飲料の味の調整にしたが、「まとめてできる処理をなるべくまとめてやりたい」は熱処理などでよくある

# 入出力例1

なるべく分岐を遅くしたい

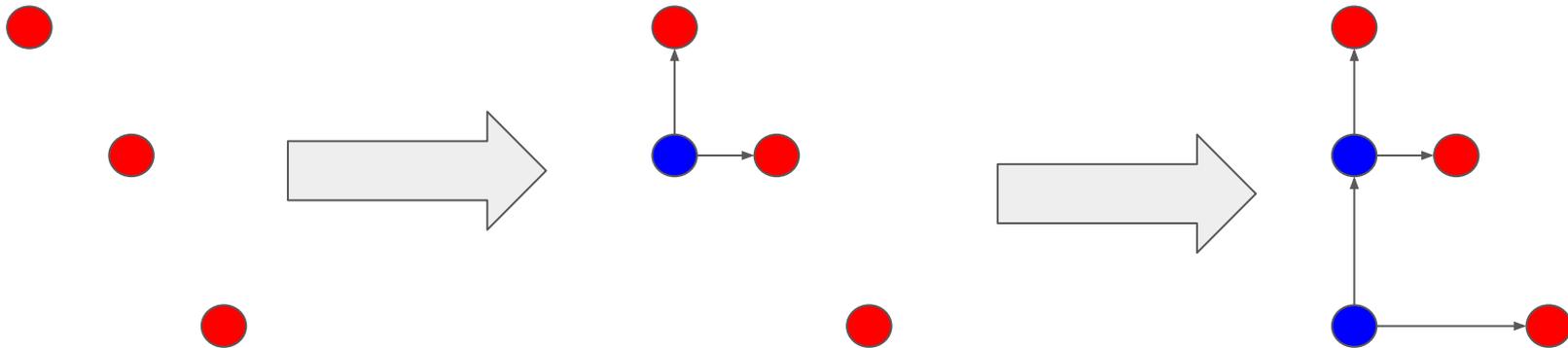


# 後ろから考える

原点から伸ばそうとすると、分岐点をどこにすればいいかが決めにくい

後ろから考えるとL字を作っていけばよさそう

→ マージするペアをいい感じに決めていく問題になる



# 貪欲法 1

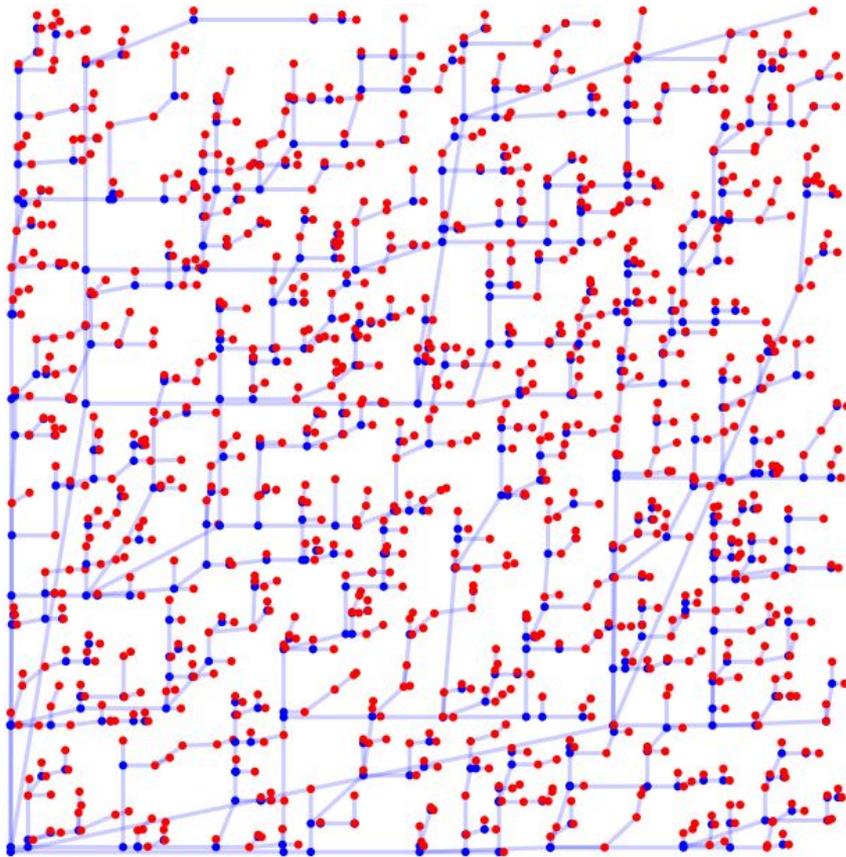
コスト最小のペアをマージする

$|x1 - x2| + |y1 - y2|$  の小さい順

3,496,265,034 (571位)

遠くに点が残って、終盤になって  
コストの大きい操作が発生する

※マージ済みの点につなぐ操作も  
許すと4.75G(170位)くらい



## 貪欲法 2

取り残される点がないように、おおよそ原点から遠い順にマージするとよさそう

原点から遠いのを優先したい  $\ni x_1+y_1+x_2+y_2$  が大きいペアを優先したい

$|x_1 - x_2| + |y_1 - y_2| - (x_1+y_1+x_2+y_2)$  が最も小さいペアをマージする

→ 5,388,249,433 (23位)

$$\begin{aligned} \text{この評価値は } & -2 * (\min(x_1, x_2) + \min(y_1, y_2)) \text{ とも書ける} \\ & |x_1 - x_2| - (x_1 + x_2) \\ & = \max(x_1, x_2) - \min(x_1, x_2) \\ & \quad - (\max(x_1, x_2) + \min(x_1, x_2)) \\ & = -2 * \min(x_1, x_2) \end{aligned}$$

# ビームサーチ

貪欲法をビームサーチにするとさらにスコアを伸ばせる

次にマージするペアの候補が  $O(N^2)$  通りあるので、よさそうなペアのみ試すなどしてビーム幅を増やす

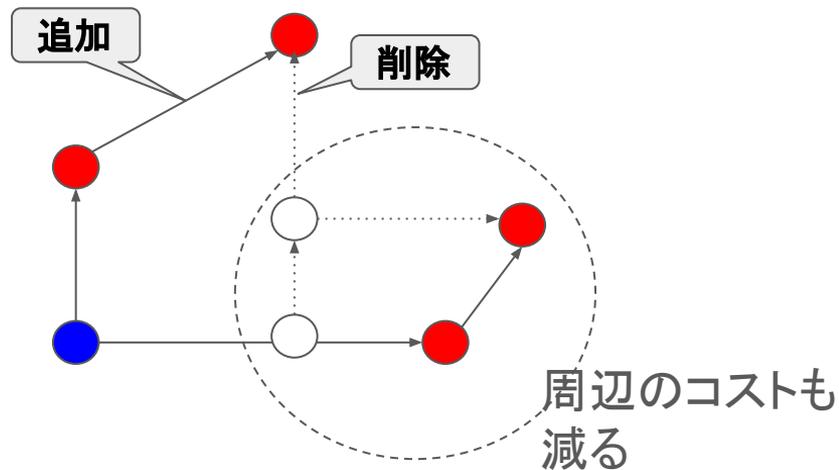
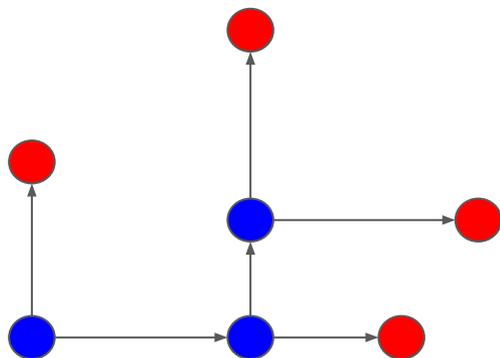
5,492,589,300 (wataさん; 本番2位相当)

# 焼きなまし

辺を消して別のところにつなぐ操作を近傍とする焼きなまし

コンテスト前日の writer 解で 5,494,335,790

本番1位(5,493,500,275)の Rafbill さんも同様の方針

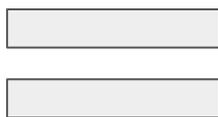
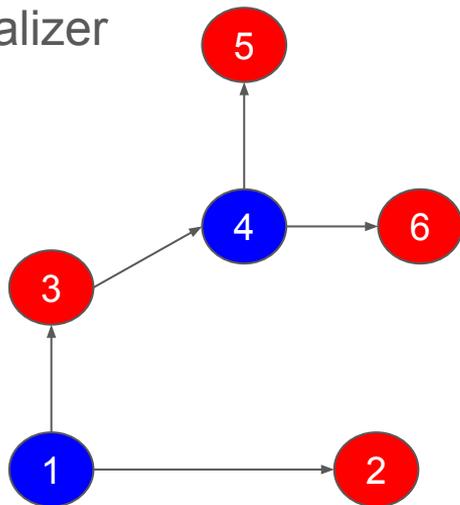


# 焼きなましの実装

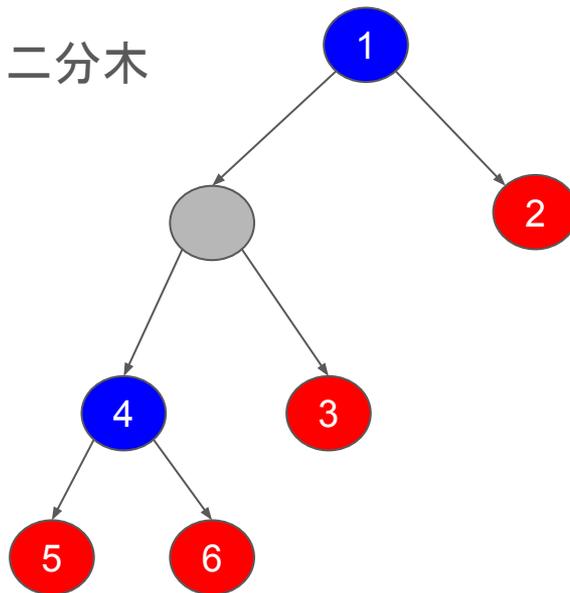
解は原点を根とし、赤い点を葉とする全二分木としてよい

左で葉ではない赤頂点 3 は全二分木による表現では二つの頂点とみなす  
木の形と葉の座標を決めると、中間ノードの座標やコストが再帰的に計算できる

visualizer



二分木



# 焼きなましの実装

部分木を切り離して別のところに移動させる(=マージする相手を変える)

差分計算などを頑張ると400万回ぐらいは回る

