

2025/3/23 AtCoder Heuristic First-step Vol.1

# 問題理解と貪欲法

松尾 充 (@terry\_u16)

# AtCoder Heuristic First-step Vol.1の目的

---

- ✓ **解法の選択方針について学ぶ**
- ✓ **貪欲法の書き方・改善方法について体験する**
- ✓ **山登り・焼きなまし法の実装を体験する**
- ✓ **今日出会ったメンバーと仲良くなる**

# AtCoder Heuristic First-step Vol.1の流れ

---

今日は基本的な貪欲・山登り・焼きなましについて学びます

約1週間のコンテストを経て、**上位チームをコンテストページで表彰**します！



# 「問題理解と貪欲法」の流れ

---

講義・グループワーク (GW)・演習を通して学んでいきます

実際に**各グループで貪欲法や改善案を考える**グループワークもありますので  
楽しみにしてください！

講義

問題文の  
理解



講義

解法選択  
方針



GW

貪欲法を  
考えよう



演習

貪欲法の  
実装



GW

改善案を  
考えよう



# 問題文の理解と解法の選択方針

# 今回のお題: Food Delivery

今回はFood DeliveryというAHC過去問を題材に皆で解いていきましょう  
ざっくり言うと「注文の効率的な配達経路を決める」問題

コンテスト時間: 2021-11-14(日) 16:00 ~ 2021-11-14(日) 20:00 (240分) AtCoderホームへ戻る

[トップ](#) [問題](#) [質問](#) [提出](#) [提出結果](#) [順位表](#) [コードテスト](#) [解説](#)

## A - Food Delivery

解説 🇯🇵 / 🇺🇸

実行時間制限: 2 sec / メモリ制限: 1024 MB

### 問題文

AtCoder社は冷めても美味しい料理をのんびり配達するフードデリバリーサービスAtCoder Foodsを運営している。このサービスでは、事前に配達の注文を沢山受け付けておき、複数の配達を一気に処理することで効率化を図っている。現在のサービスエリアは二次元平面上の正方形領域  $\{(x, y) \mid 0 \leq x, y \leq 800\}$  で表され、中心の  $(400, 400)$  の地点にAtCoder社のオフィスがある。今日は注文が1000件入っており、 $i$  ( $1 \leq i \leq 1000$ ) 番目の注文は  $(a_i, b_i)$  にあるレストランから  $(c_i, d_i)$  への配達依頼である。

配達員の一人である高橋くんの今日のノルマは50件の注文を処理することである。高橋くんは1000件の注文の中からちょうど50件の部分集合  $S \subseteq \{1, \dots, 1000\}$  を自由に選び、以下の条件を満たすルート  $(x_1, y_1), \dots, (x_n, y_n)$  で配達を行う。

- 各  $i \in S$  に対して、 $(a_i, b_i)$  を訪れた後に、 $(c_i, d_i)$  を訪れる。すなわち、 $(x_s, y_s) = (a_i, b_i)$ 、 $(x_t, y_t) = (c_i, d_i)$ 、 $s < t$  となる整数組  $(s, t)$  が存在する。
- $(x_1, y_1) = (x_n, y_n) = (400, 400)$ 。

あるレストランで料理をピックアップした後、その料理を配達先に届ける前に別のレストランで料理をピックアップしたり、別の料理を先に届けても良い。高橋くんは力持ちなので任意個の料理を同時に運ぶことが可能である。

$(x_i, y_i)$  から  $(x_{i+1}, y_{i+1})$  への移動にはマンハッタン距離  $|x_i - x_{i+1}| + |y_i - y_{i+1}|$  の時間がかかり、配達ルートの総移動時間は  $T = \sum_{i=1}^{n-1} |x_i - x_{i+1}| + |y_i - y_{i+1}|$  となる。総移動時間が出来るだけ短くなるように担当する注文  $S$  の選択と配達ルートを最適化して欲しい。

AHC過去問なので  
検索すれば解説記事が出てきますが、  
コンテスト期間中はできるだけ  
解説記事などは見ずに自分の力で  
チャレンジしてみてください！

# Food Delivery概要

---

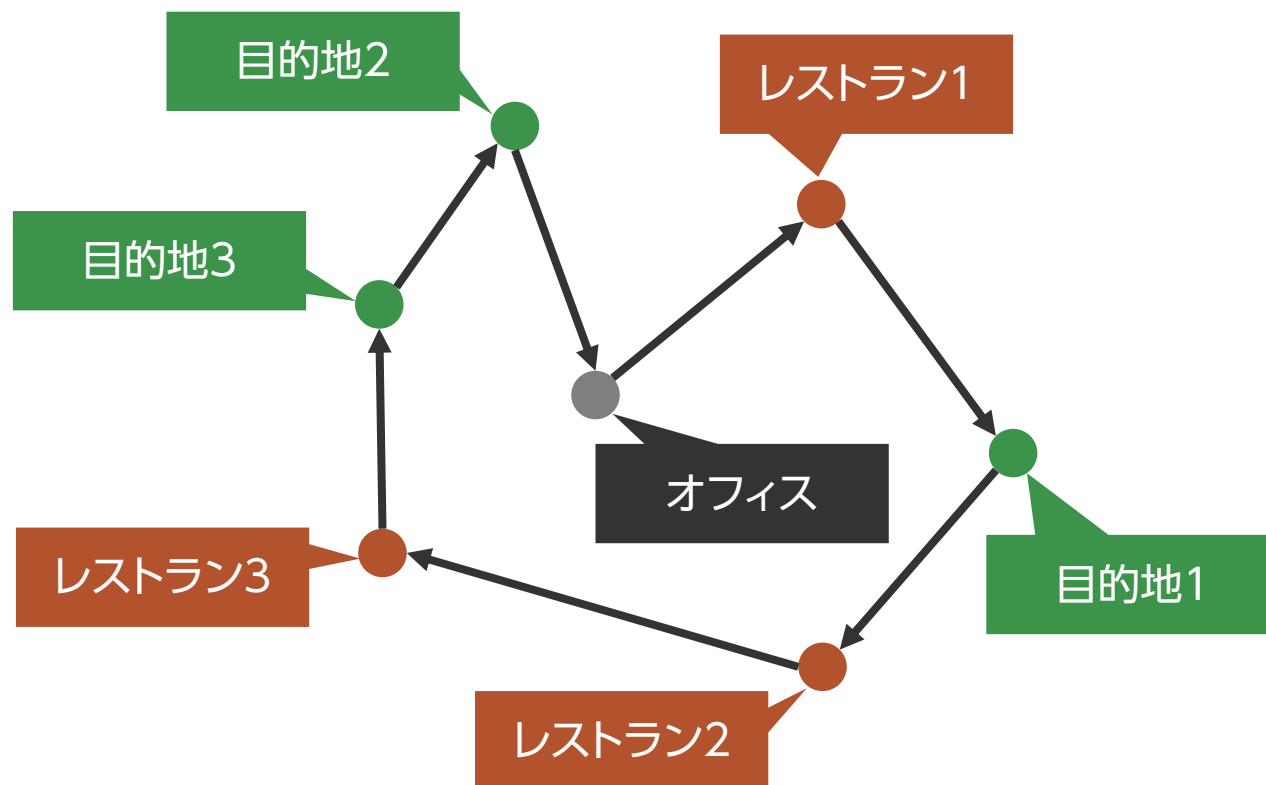
ざっくりまとめると以下のような問題

- 二次元平面が舞台
- 1000件の注文が入っている
- 注文  $i$  はレストラン  $(a_i, b_i)$  から目的地  $(c_i, d_i)$  への配達
- **1000件の中から好きな50件を選んで配達を行う**
- 料理は複数個同時に運ぶことができる
- 2点間  $(x_i, y_i), (x_{i+1}, y_{i+1})$  の移動にはマンハッタン距離  $|x_i - x_{i+1}| + |y_i - y_{i+1}|$  だけの時間がかかる
- **オフィスから出発し、料理の配達を終えてオフィスに戻ってくるまでの総移動時間をできるだけ短くしたい**

# ビジュアライザ確認

問題文を読むのとビジュアライザの解答例を見るのは合わせて1セット

最近のAHCは例が付いているので、眺めて問題文を正しく理解しよう



1. オフィス発
2. レストラン1
3. 目的地1
4. レストラン2
5. レストラン3
6. 目的地3
7. 目的地2
8. オフィス着



## 解法選択フローチャート

---

問題文を読む

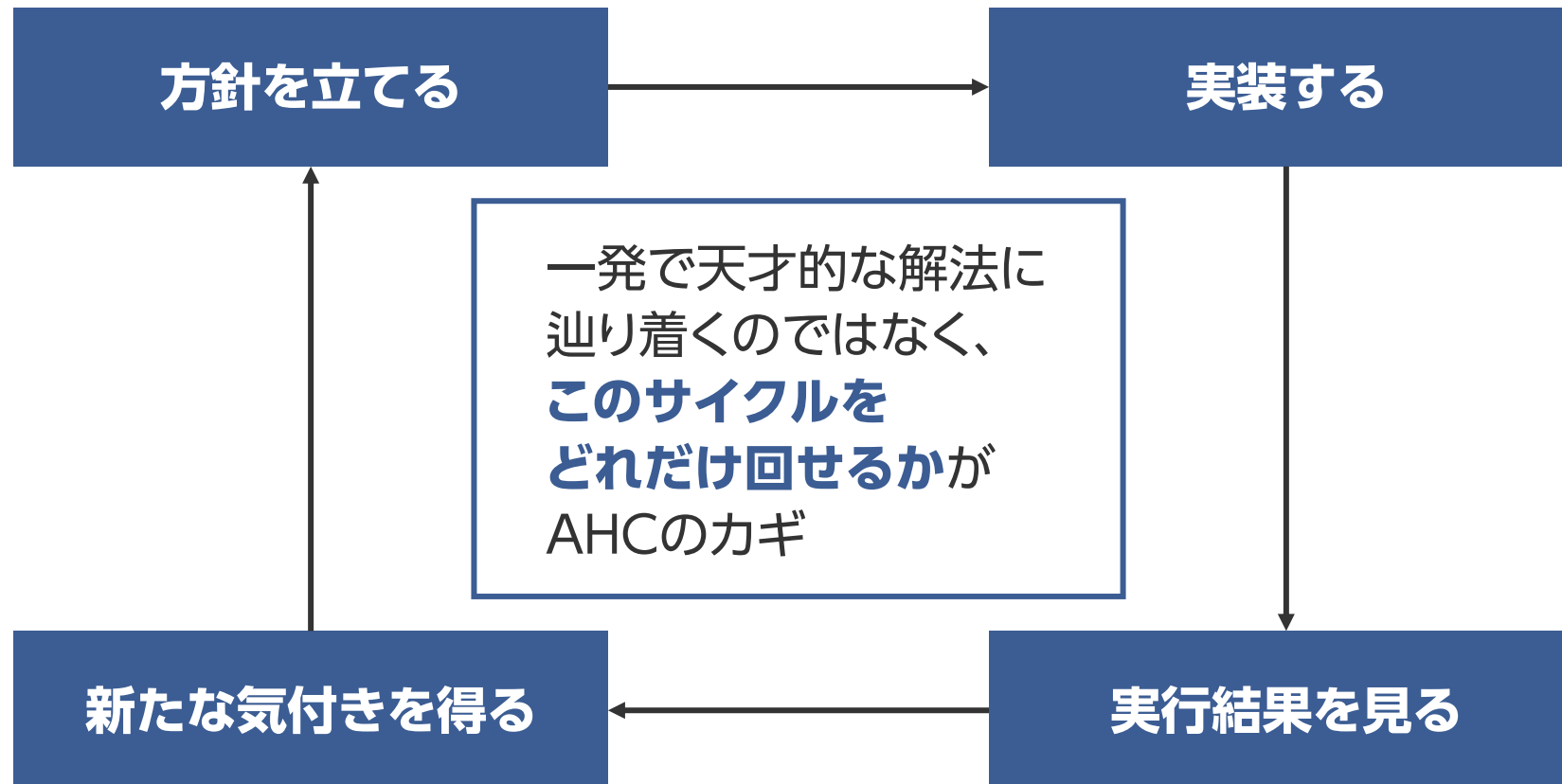


```
graph TD; A([問題文を読む]) --> B[普通の貪欲を書く];
```

**普通の貪欲を書く**

# なぜ普通の貪欲？

問題文を一度読んだだけで何が重要か理解するのは**常人には不可能**  
**簡単な解法を動かしてみる**ことで、問題を**より深く理解**できる



# 貪欲を書くときに重要なこと

いきなり**天才貪欲**を書こうとしないことが大事

問題文の理解と割り切り、**5分で思い付く超簡単な貪欲**から始めよう

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
72	73	74	75	76	77	78	79										


## TOYOTA Programming Contest 2023 Summer final

「コンテナがランダムな順番で運び込まれるので  
できるだけ**順番通りに運び出せるように配置**してください  
ただし入口から通路を通して辿り着けないマスは操作できません」  
という問題

自分は初手で「**運び込むときは適当なところに配置し  
運び出すときは到達可能で最も番号が小さいコンテナを運び出す**」  
という貪欲を実装

「**とりあえず簡単な貪欲**」は上級者でも有効な戦法

ちなみにこのあと優勝しました

貪欲法

## 貪欲を考えてみよう

---

同じグループの人と一緒に**どんな貪欲が考えられるか相談**してみましよう！  
**2分間**個人で考えてみて、**3分間**で案を出し合ってみてください！

### 恥ずかしがらない！

5分どころか5秒で考えた“あたりまえ貪欲”でOK！解法が被っても気にしない！

### 色々な考え方を認める！

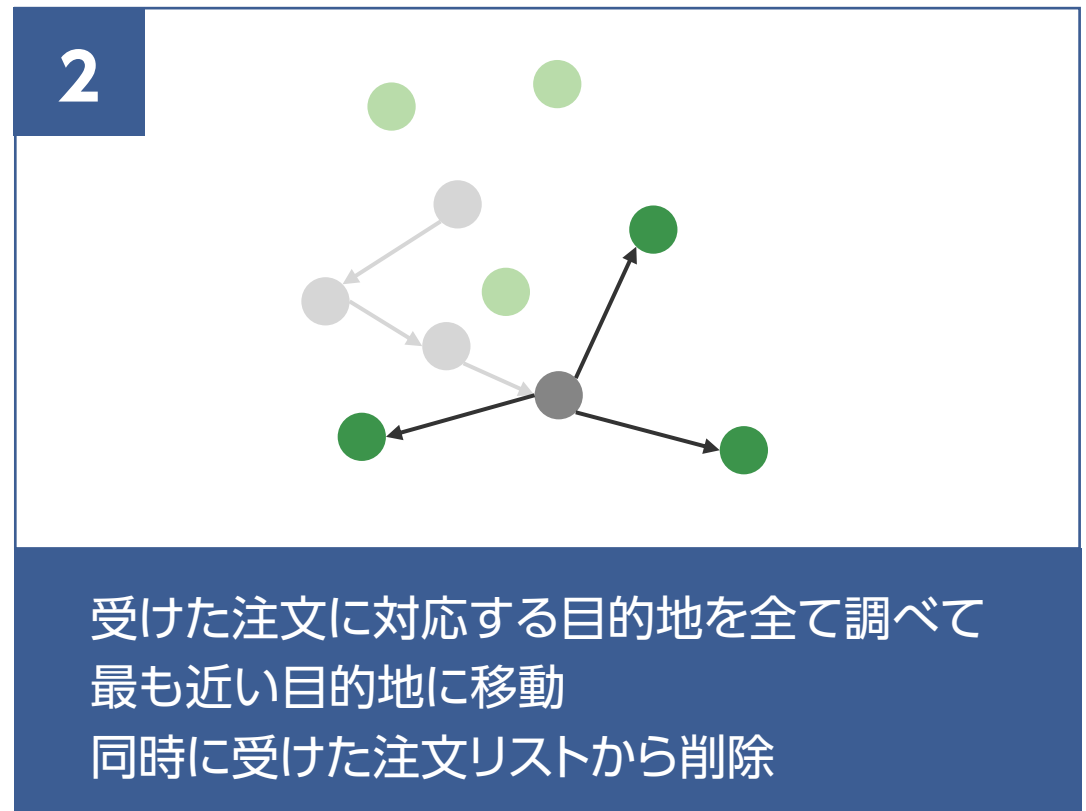
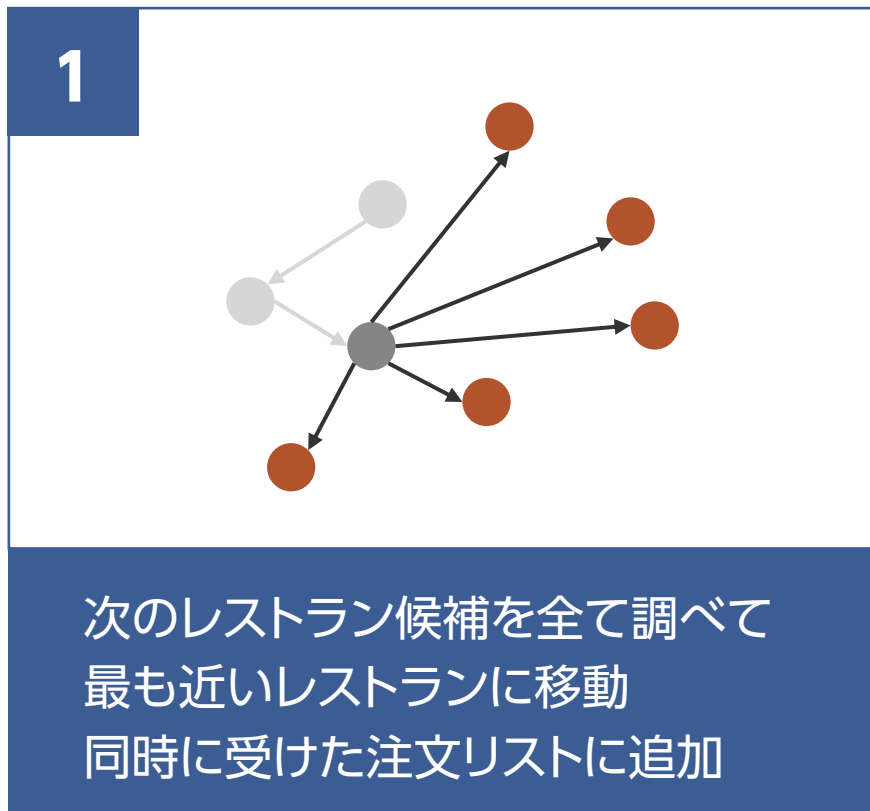
最適化には“多様性”が大事！自分になかった視点を褒めてあげよう！

### アイデアを膨らませる！

違う視点から見たらもっと良くなるかも？相手を尊重しつつさらに伸ばしてみよう！

# 貪欲法の一例

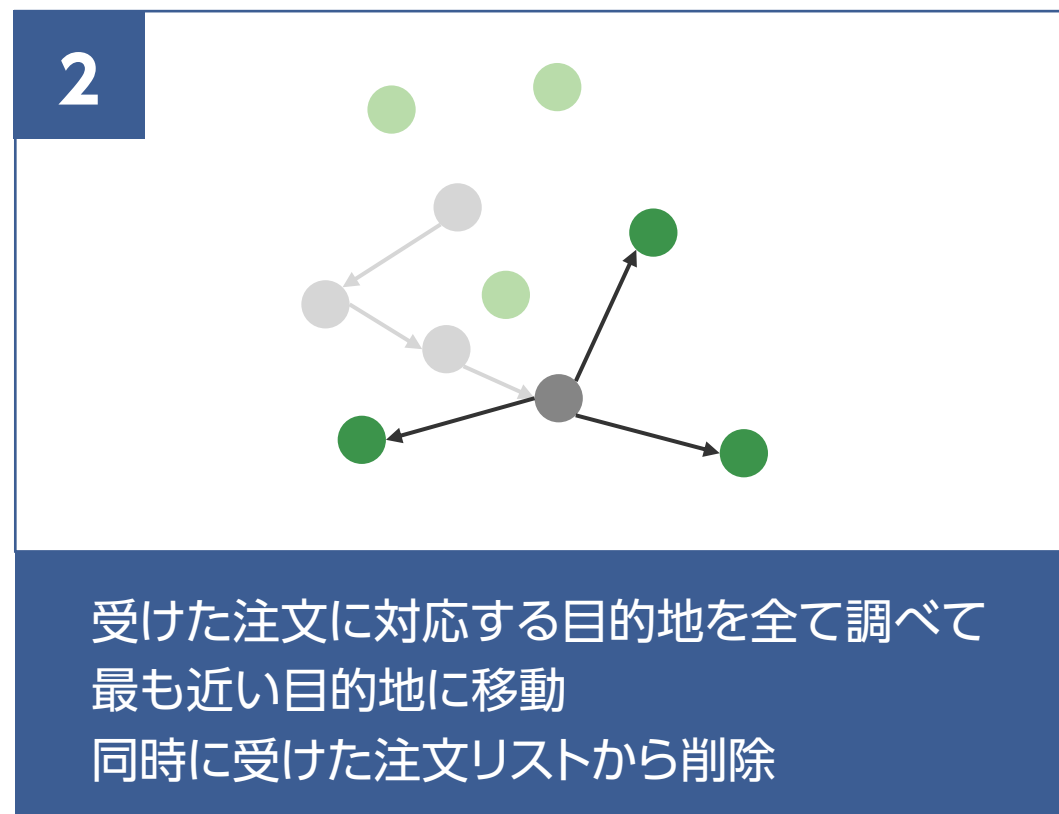
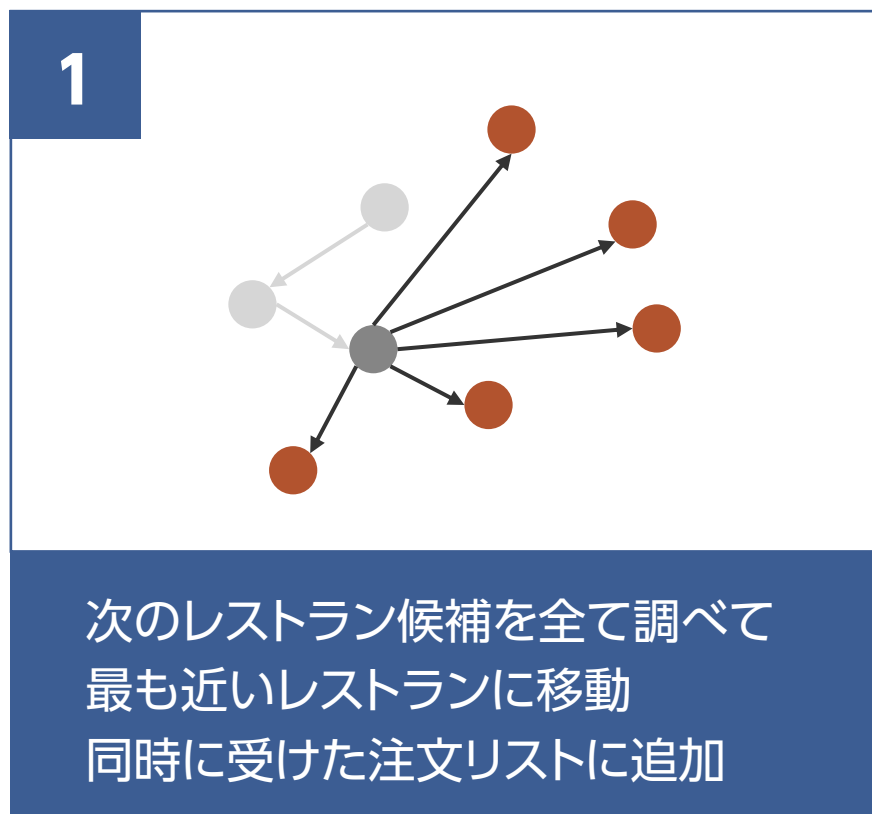
1. 現在地から最も近いレストランに行くことを50回繰り返す
2. 一番近い目的地に行くことを50回繰り返して、オフィスに戻る



# 貪欲法を実装してみよう

テンプレコードを配布するので、穴埋め形式で貪欲法を実装してみましょう

**1段階目を実装したら一旦実行してビジュアライザを見る**のがオススメ



# 実装タイム

困ったときには**問題ページの質問タブ**から受け付けます！  
必要事項をご記入頂ければスタッフが順番に伺います

[🏠 トップ](#) [❓ 質問](#) [📊 順位表](#)

## 質問

質問がまだありません。

[質問する](#)



## 必要事項

- ① テーブル名
- ② 言語
- ③ 質問内容

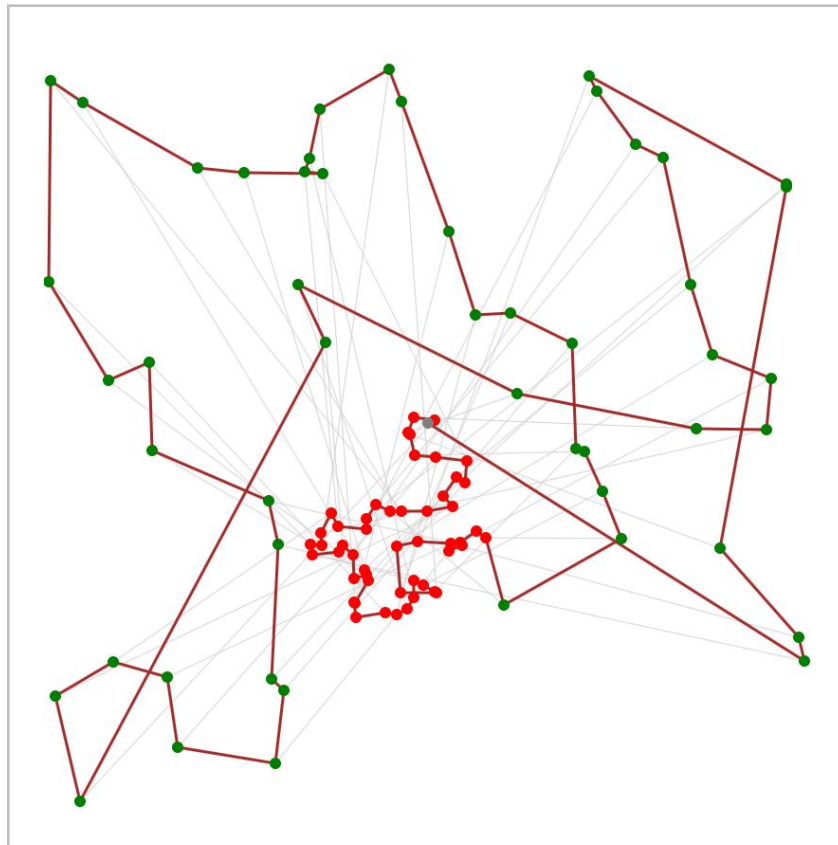
順番待ちになるかもしれませんが  
よろしくお願いします



# ダメ出しをしてみよう

---

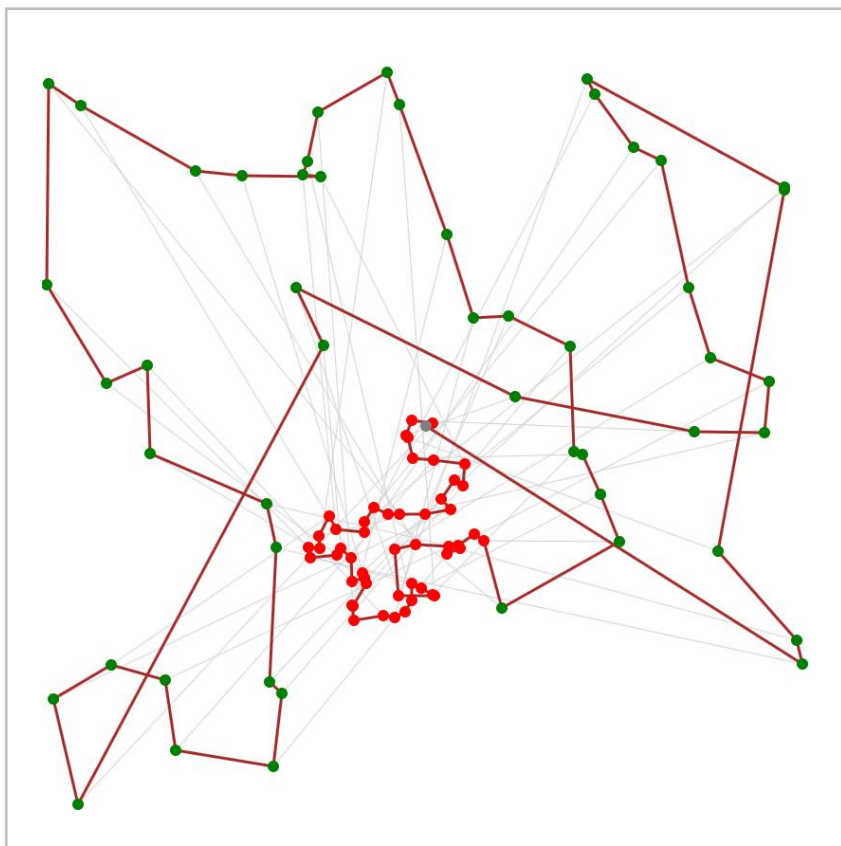
ビジュアライザを見て、**イマイチな点**と**シンプルな改善案**を挙げてみましょう  
グループで**3分**ほど話し合ってみてください！



場当たりの改善案でもOK！  
直感にしたがって  
どんどん挙げてみましょう！

# ダメ出しの例

例としてこのようなイマイチな点を挙げてみました  
この中の1つについて**実際に改善を行ってみましょう**

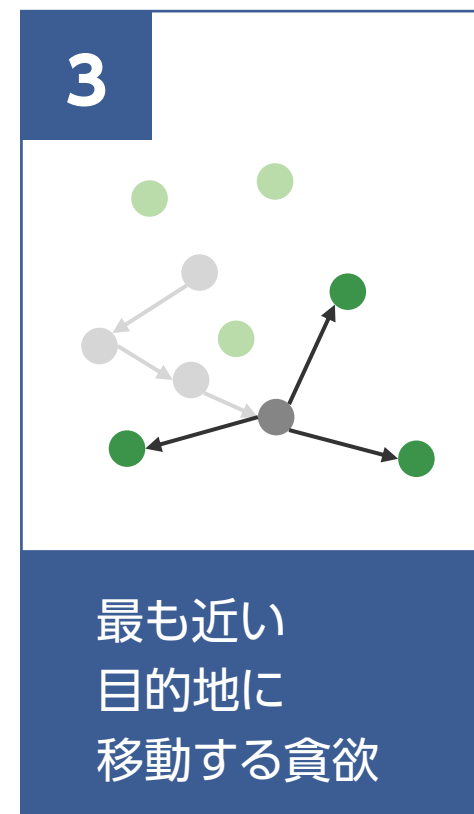
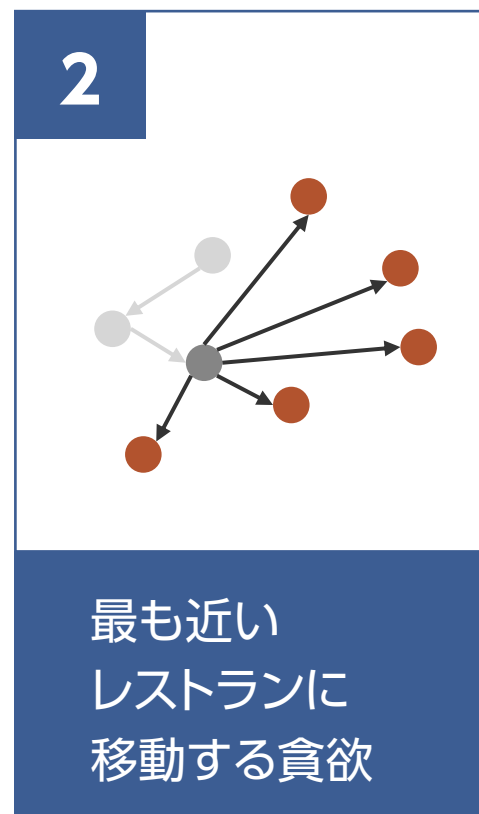
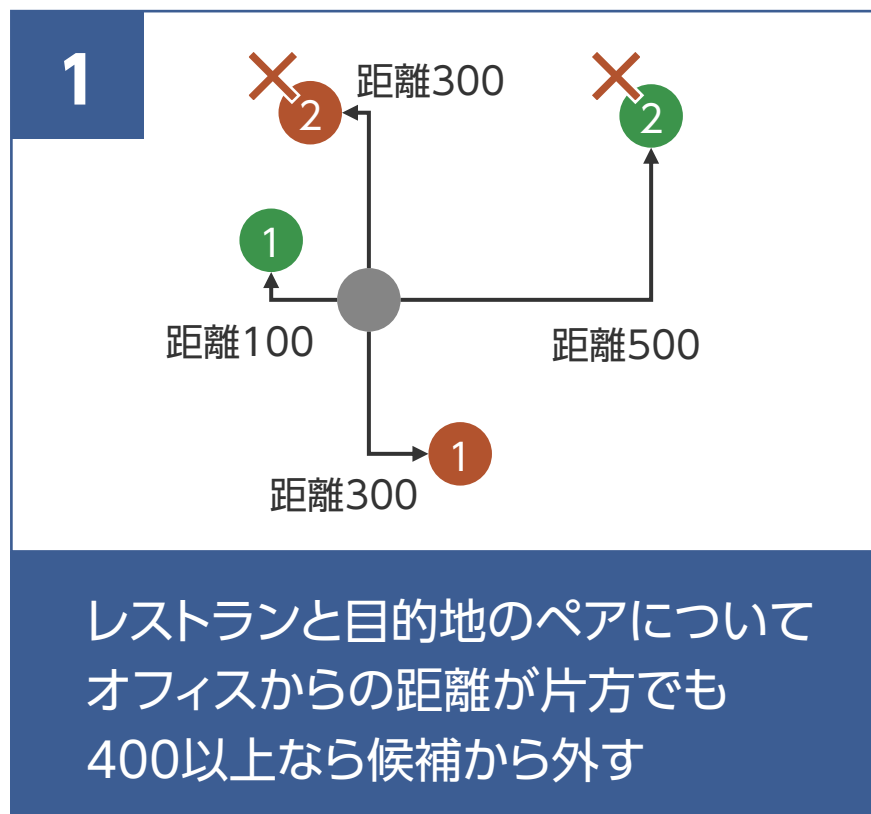


- レストランの近さだけを見ているので**目的地が全体的に遠すぎ**
- 後半のルートが**行ったり来たり**で効率が悪い
- レストランを回る**ついでに目的地に配達**しても良いのでは
- レストラン-目的地の**ペアの距離** (灰色の線の長さ) **が遠すぎる**

# 貪欲法の改善

遠い目的地に行くのはよくなさそうなので

**レストラン・配達先のどちらもオフィスに近い注文だけを対象**にしてみる

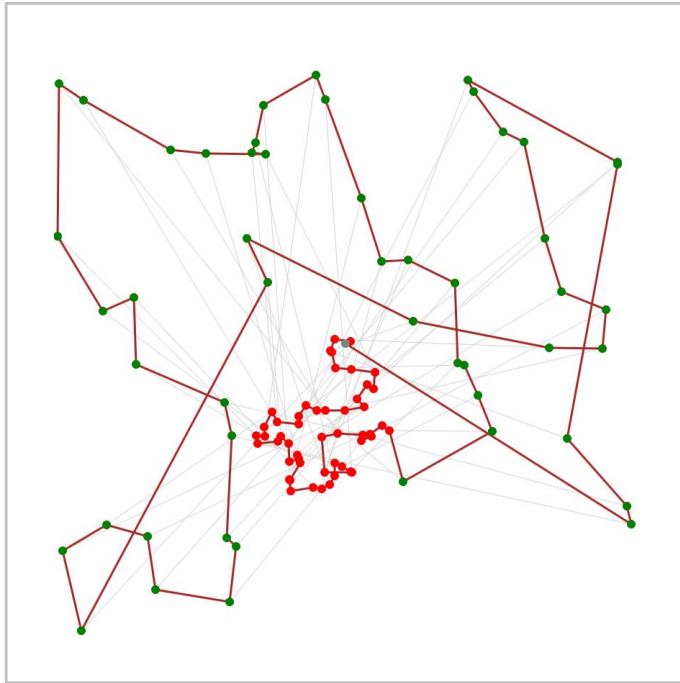


# 改善結果

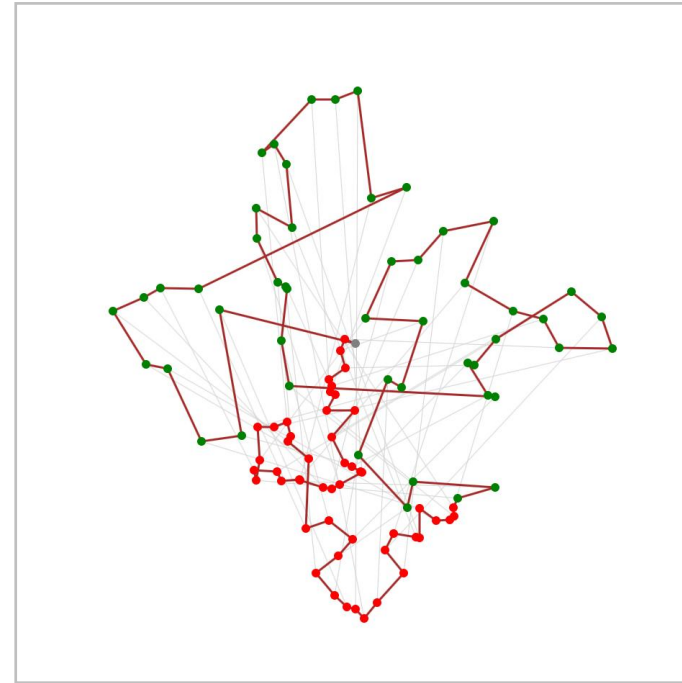
---

改善結果はこんな感じ

まだまだ改善余地があるので、色々と試してみてください！



総移動距離 **7918**



総移動距離 **6308**

**ご清聴ありがとうございました！**