

ICPC2021 模擬地区予選 G 問題

Pizza Delivery

原案: smiken

問題文: tsutaj

データセット: riantkb

解答: beet, hos, riantkb, tsutaj

解説: tsutaj

2022 年 3 月 6 日

Pizza Delivery

- ▶ N 人の客が同時にピザの注文を行った
 - ▶ t_i : ピザ屋から客 i の家までの所要時間
 - ▶ 客 i の家からピザ屋まで帰るときも同様
 - ▶ a_i : 客 i の怒りっぽさ
- ▶ これらの注文を一人の配達員が捌いていく
- ▶ 客は配達時間が遅くなるほど・先に配達された別の客の数が多いほど、ストレスがたまる
 - ▶ 問題文の $s_i = a_i \times (h_i + p_i)$ に相当する部分
 - ▶ p_i : 客 i に配達する前に配達された客の人数
- ▶ N 人の客のストレス量の合計の最小値はいくつになるか？

- ▶ N 人の客を適切に順序付けて、その順番通りに配達していく問題
- ▶ “ $dp[S] :=$ 注文を届け終えた客の集合が S であるときの、ストレス量の合計の最小値” という bitDP が考えられる
 - ▶ しかし $O(2^N)$ かかるので間に合わない
- ▶ bitDP では客の順序を全て考慮することができるが、本当に全部の順序を考えなければならないのか？
 - ▶ もう少し楽をできないか考えてみよう

$N = 2$ の場合

- ▶ 簡単のため、客が 2 人しかいない場合をまず考える
 - ▶ 以降、客のストレス量の合計を X で表す
 - ▶ X の下付き文字は客の順序を表す
- ▶ 客 1 → 客 2 の順で配達を行う場合
 - ▶ 客 2 への配達時間が $2t_1 + t_2$ であることに注意
 - ▶ 客 2 より前に客 1 への配達完了している (その分のストレスも加算)

$$X_{1 \rightarrow 2} = a_1 t_1 + a_2 (2t_1 + t_2 + 1) \quad (1)$$

- ▶ 客 2 → 客 1 の順で配達を行う場合
 - ▶ 客 1 への配達時間が $2t_2 + t_1$ であることに注意
 - ▶ 客 1 より前に客 2 への配達完了している (その分のストレスも加算)

$$X_{2 \rightarrow 1} = a_2 t_2 + a_1 (2t_2 + t_1 + 1) \quad (2)$$

$N = 2$ の場合

- ▶ どちらの方がストレス量が少ないか知りたいので、大小関係を見よう
- ▶ 引き算を試みる
 - ▶ 次の式の結果が負なら $X_{1 \rightarrow 2}$ のほうが、正なら $X_{2 \rightarrow 1}$ のほうがストレス量が少ない
 - ▶ 結果が 0 ならストレス量は等しい (以降、“0 以上” と “負” に分類することにする)

$$\begin{aligned}X_{1 \rightarrow 2} - X_{2 \rightarrow 1} &= \{a_1 t_1 + a_2 (2t_1 + t_2 + 1)\} - \{a_2 t_2 + a_1 (2t_2 + t_1 + 1)\} \\ &= 2a_2 t_1 + a_2 - 2a_1 t_2 - a_1 \\ &= a_2 (2t_1 + 1) - a_1 (2t_2 + 1)\end{aligned}$$

- ▶ ストレス量の大小を知りたいので、上式の符号に興味がある
 - ▶ $a_2 (2t_1 + 1) - a_1 (2t_2 + 1)$ が 0 以上になるかならないか
- ▶ 符号に応じて、適切な順序を選び取ればよい

$N = 2$ の場合

- ▶ 前ページより、2 人の客の順序は以下の比較関数によって決定できる

比較関数

- ▶ 2 人の客 u, v に対する比較関数 $C(u, v) := a_v(2t_u + 1) - a_u(2t_v + 1)$
- ▶ $C(u, v) < 0$ のとき、 $u \rightarrow v$ という順序で配達すると最適
 - ▶ このとき、移項して $\frac{a_v}{2t_v+1} < \frac{a_u}{2t_u+1}$ が成り立つ…(★)
- ▶ $C(u, v) \geq 0$ のとき、 $v \rightarrow u$ という順序で配達すると最適

$N \geq 3$ の場合 (一般の場合)

- ▶ 実は 3 人以上いる場合でも同様に順序を決めて良い
 - ▶ 客 u, v について最適な順序が $u \rightarrow v$ で、客 v, w について最適な順序が $v \rightarrow w$ であるとする
 - ▶ このとき、客 u, w について $u \rightarrow w$ が最適な順序になる
 - ▶ 前ページ (★) 式を使って導かれます
 - ▶ これを全ての客に適用すると、 $\frac{a_i}{2t_i+1}$ の降順に並べると最適になる！
- ▶ 例えば C++ の場合、比較関数によるソートは以下のように書けます
 - ▶ 小数演算が入らないよう、掛け算で処理するのをおすすめします

```
// vector<long long int> T(N), A(N) が上で定義されている
vector<int> order(N);
// 0 から N-1 まで連番でいれる
iota(order.begin(), order.end(), 0);
// 比較関数でソート
sort(order.begin(), order.end(), [&](int i, int j) {
    return A[i] * (2 * T[j] + 1) > A[j] * (2 * T[i] + 1);
});
```

▶ Writer 解

- ▶ beet (C++・97 行・999 bytes)
- ▶ hos (C++・61 行・1463 bytes)
- ▶ riantkb (Python・25 行・428 bytes)
- ▶ tsutaj (C++・31 行・705 bytes)

▶ 統計

- ▶ AC / tried: 29 / 36 (80.6 %)
- ▶ First AC
 - ▶ The University of Tokyo: DELIAIR (17 min 53 sec)