第 35 回競技部門：30058

タイトル：寿司が好きなコンピュータ，何か知ってる？
「データ巻き」ってね

学校名：神戸市立工業高等専門学校

1. はじめに

課題の型抜きパズルは，抜き型の種類や使用位置など，分岐が非常に多い．そこで，後述する基本の動きを一般化し，そこから考えられる最適解を見つけるような戦略を取る．

2. アプローチ

2.1 大まかな流れ

基本の動きでは，主に行と列を別々に揃えていく戦略にした．その流れを以下に示す．

(1) 最終盤面を下の行から順に走査し，各行の 0-3 の各要素数を揃える
   (a) 縦方向の寄せのみで過剰な要素を不足した要素で置換する
   (b) 残った過剰な要素のピースを，そのピース近隣列の不足した要素で置換する
(2) 最終盤面を右列から順に走査し，各列を横方向の寄せのみで揃える

2.2 ピースの移動法

ピースの移動は，目標位置までに存在するピースを抜いて移動させる．定型抜き型のみの使用では，移動させたい距離をNと置くと必要手数は$O(\log N)$手となる．一方，盤面端に移動させるだけの場合は，移動させたいピースを抜いて移動させると，1 手で済む． これにより，一番下の行・一番右の列に揃え，反対側に移動させることで，より少ない手数で盤面を揃えられる．

また，(1.b)では，揃えている行と平行に移動させる必要がある．盤面端に揃えることは難しいため，先述の$O(\log N)$手で揃える． しかし，揃えている途中の一番下の行を回避して型抜きする必要があるため，定型抜き型のタイプ II を用いて，盤面の崩れを防ぐ．

2.3 複数パターンの流れ

前述の流れは，下の行から右の列という順番になっている．しかし，走査の始点と終点で 8 通りのパターンがあるため，全て検証し，最適な流れを見つける．このために，初めから盤面をそれぞれ反転させた状態で同様の流れを並列で処理する．

2.4 複数種の抜き型の使用

課題では，適切な抜き型の使用が手数省略の最大のカギである．そこで，盤面の崩れない定型抜き型を使用した際の過剰な要素数などから各評価値を判断する．各評価値を比較し，最適な抜き型の種類を決定する．

3. システム

全体を JavaScript で実装し，ソルバと UI で実装を分割している．

3.1 ソルバ

Promise を利用した非同期処理をベースに，WebWorker で並列化している．

具体的には，サーバ・UI との通信，アルゴリズムの実行をマイクロサービスとして分割している．プログラム全体をステートマシンとして構築して，サービス間で状態を共有できる仕組みにしている．また，一方向に送受信可能なデータチャネルを用いて相互に通信を行えるようにしている．

実行時は，最適化された単独バイナリとして実行できるようにビルドする．

3.2 UI

React を利用し，サーバからのイベントストリームを受け取り，ソルバの進捗をリアルタイムで表示する． ソルバと完全に分離することで，長時間の計算でも UI がブロックされない．