

部 門	競 技 部 門	No.1 登録番号	
-----	---------	-----------	--

No.2

1) 予定開発期間：7ヶ月

2) 予定開発人数：3人

	4月	5月	6月	7月	8月	9月	10月
問題分析	←→						
設計	←→						
実装		←→					
試用・トレーニング					←→		

No.3	<p>実現方法</p> <p>1) フィールド上のペアを多く作るアルゴリズム</p> <p>今回の問題ではペアを全て揃えることができる。図1に示すように、まず盤面の上から2行を対象として処理を行い、その後右側の2列を順に揃えていく。この操作を下方および左方向に繰り返していくことで、最終的にすべてのペアを接続できるよう構成されている。また、四隅から揃える方法も見えている。四隅のうち一番少ない手数で揃えられるペアを優先的に揃えている。</p> <p>さらに、4×4の領域に対しては、すべての可能な配置に対して幅優先探索による全数探索を行い、それぞれの状態からの最短経路を求める。この探索によって得られた最短手のデータはデータベースとして構築されており、任意の4×4の部分盤面が出現した際に、瞬時に最適な手順を参照できるようになっている。このデータベースを活用することで、局所領域における操作効率を飛躍的に高めることが可能となった。</p>
	<p>2) 導きの手数を少なくする工夫</p> <p>このプログラムは1つ1ペアを作成していくが、基本的にマス揃えていく順番、ペアの向きは固定にしている。ペアを揃えていく際には、既にそろえたところを回転することがないような揃え方で揃えてくのだが、ペアの揃え方を固定にすることによってi番目のペアをそろえようとしたとき、既にそろっているところが確定する。そのため一度揃えたところを回転させないようにi番目のペアを揃える手順を1回計算してしまえばその結果は使いまわすことができる。よって、その結果をファイルに保存し、それを読み込むことによって毎回計算する必要がなくなり、大幅に1回あたりの計算時間を削減できる。</p> <p>また、揃え方を5つほど用意しておくことによって、ペアを揃える順番は固定だが、揃えるときの経路を探索することができビームサーチを使うことで、総合的に手数が少なくなるような解を探索している。1つの状態においての計算時間を大幅に削減できたことによって、ビーム幅を2000ほどまで増やしても最大サイズの24の時に3分くらいで実行できる。ビームサーチを行うときは値を <code>priority_queue</code> に追加すること、ポインタを管理するラッパークラスを追加することによって少しでも高速にしている。1通りしか試さなかったときはサイズが24で750回程度かかったのがビームサーチを用いると530回程度と1ペア当たりにかかる手数を平均2回未満で揃えることができる。</p> <p>基本的には揃える順番は固定だが、辺を揃える順番は少しだけ変えることができ手数を削減できる。プログラムの内部的には全体を回転させるという処理を数回繰り返し、揃える辺が左上になるように扱うことで1度求めた結果に落とし込める。ただし、実際は全体を回転させる処理は不要なため、座標を計算し、全体を回転せずに同じ動きを実現している。</p>
	<p>3) その他 (独創的なところ)</p> <p>4×4のフィールドとそれに対応する導きをLMDBに保存する際、4×4のフィールドを単純な方法で表現すると、フィールドには8種類の数字が配置されるため、各マスを$\lceil \log_2(8) \rceil = 3$ビットで表現し、全体で$3 \times 16 = 48$ビットが必要となる。</p> <p>しかし、フィールドの数字を左上から順に再割り当てすることで、左上のマスは必ず0、その右隣のマスは0または1になるなど、出現する数字に偏りが生まれる。この偏りを利用することで、盤面を34ビットで表現できるようにした。また、これにより表現可能な盤面の総数も削減している。</p> <p>対応する導きについては、すべての手順を保存するのではなく、盤面ごとに一つのみとし、全体を木構造として記録している。</p>

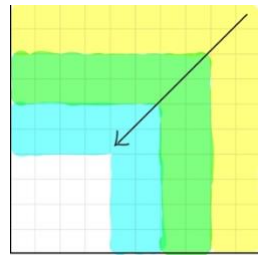


図 1 ペアの揃え方

No.4	<p>開発環境</p> <p>Visual Studio Code</p> <p>Vim</p> <p>開発言語</p> <p>C++</p>
------	---