

部 門	競 技 部 門	No.1 登録番号	30010
-----	---------	-----------	-------

No.2	1) 予定開発期間：6ヶ月 2) 予定開発人数：3人																																								
	<table border="1"> <thead> <tr> <th></th> <th>4月</th> <th>5月</th> <th>6月</th> <th>7月</th> <th>8月</th> <th>9月</th> <th>10月</th> </tr> </thead> <tbody> <tr> <td>問題分析</td> <td colspan="2" style="text-align: center;">←————→</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>設計</td> <td></td> <td colspan="4" style="text-align: center;">←————→</td> <td></td> <td></td> </tr> <tr> <td>実装</td> <td></td> <td colspan="5" style="text-align: center;">←————→</td> <td></td> </tr> <tr> <td>試用・トレーニング</td> <td></td> <td></td> <td></td> <td></td> <td colspan="3" style="text-align: center;">←————→</td> </tr> </tbody> </table>		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					←————→		
		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.2** 実現方法

1) フィールド上のペアを多く作るアルゴリズム

まず、盤面の下端2行を左から順に横並びのペアで埋める。次に、既にペアとなった下端2行を除く盤面について、右端の2列を下から順に縦並びのペアで埋める。ペアの作成順序を図1に示す。この図では、説明のため、エンティティの値を、ペアを成立させる順に対応させている。実際にはエンティティの値とは無関係にペアを成立させる。下端のそれぞれのペアを作る方法には、ペアの左側のエンティティに対応するエンティティを右隣に持ってくる操作とペアを配置したい場所と同じ行の右側で縦に成立したペアを回転移動し配置する操作の2種類がある。これらの方法を2)で述べるビームサーチにより組み合わせる。ただし、前者の方法は、下端2行の最も右のペアには使えない。右端の2列を揃える方法については、下端2行の場合と同様である。これらの操作が完了すると、盤面の下端2行と右端2列がすべてペアで埋まっているため、一つ小さいサイズの盤面であるとみなせる。これを盤面の一边の長さが4になるまで繰り返す。一边の長さが4の盤面については、幅優先探索ですべてのペアを揃える。

				9	7
				9	7
				8	6
				8	6
3	3	4	4	5	5
0	0	1	1	2	2

図1 下端及び右端のペア成立順序

**No.3** 2) 導きの手数を少なくする工夫

手数を少なくするために、ビームサーチを用いる。ビームサーチでは、与えられた盤面を初期状態とし、そこから1)で述べたペア成立のための2種類の操作を適用した盤面を生成する。生成した盤面について、現時点で下端2行及び右端2列で揃っているペアの数を評価値として枝刈りを行う。枝刈り後のそれぞれの盤面について、同様に操作適用後の盤面の生成と枝刈りを繰り返す。これを下端2行と右端2列がそろった盤面が生成されるまで行う。図2にビームサーチによる探索の例を示す。図2では、各円がそれぞれの状態を表しており、円内の数値が評価値である。また、各列がある時点で保持している全ての状態を表している。左の列から順に、操作を適用した盤面の列挙と、枝刈りを繰り返している。この例では、ビーム幅が3に設定されているため、生成した盤面から評価値の高い順に3つの盤面を残し、次の探索を行っている。

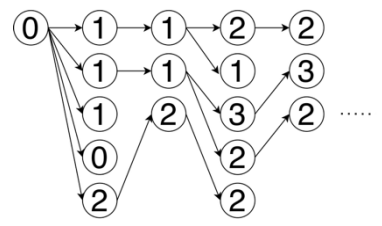


図2 ビームサーチによる探索

3) その他 (独創的などころ)

使用する3台のPC間にTCPコネクションを構築し、探索状態など必要な情報を共有する。また、直近に受理された解答の操作回数を保持するトークンを3台のPCでリング上に受け渡し、常により少ない操作回数の解答のみを自動で送信できるようにする。コネクションの状態はプログラムが常に監視し、通信エラーやその他問題を検知するとログを表示し、解答の送信を手動で行えるようなCLIを提供する。

競技用プログラムは、主にC++を用いて実装する。並列計算のための独自のスレッドプールユーティリティを作成し、利用する。また、NVIDIAのGPUを搭載したWindows PCでは、CUDAを用いることで高速化を図る。

**No.4** 開発環境  
 プログラミング言語: C++23, Python3.13  
 OS: Windows 11, macOS Sequoia, Ubuntu 24.04.2 LTS  
 エディタ: Visual Studio Code  
 HTTP 通信ライブラリ: cpp-httplib  
 JSON ライブラリ: nlohmann/json  
 GPGPU ライブラリ: CUDA 12.8