

AHC050 解説

physics0523

問題概要

$N \times N$ のグリッドがある。このグリッドのうち M マスには岩が置かれており、他のマスには何も置かれていない。

あなたは、残りの $N \times N - M$ マスに岩を置く順番 $P = (P_0, P_1, \dots, P_{N^2-M-1})$ を指定する。

この指定を行った後、次ページのゲームを行う。一度ゲームが始まると、 P を変えることはできない。

問題概要

- ▶ ロボットが空きマスを一様ランダムに選び出現する。
- ▶ $i = 0, 1, \dots, N^2 - M - 1$ について以下を繰り返す。
 - ▶ まず、ロボットが移動する。ロボットは上下左右の4方向を一様ランダムに選択し、その方向に岩にぶつかるまで直進する。
 - ▶ 次に、マス P_i に岩を置く。
 - ▶ マス P_i にロボットがいた場合、ロボットは潰されゲームが終了する。
 - ▶ この時点でロボットが潰されていない場合、賞金 1 円を得る。

うまく列 P を定め、獲得賞金の期待値を最大化せよ。

$$N = 40, N^2/10 \leq M \leq N^2/4$$

解法1

まずはスコアを気にせず、正当な解をひとつ出力することを目指す

- ▶ マス (i, j) を i と j に関する2重ループで調べ、空きマスなら岩を置く (列 P の末尾に追加する) ことを繰り返す

39,025,779 点 (満点の 26.01%, 1058位相当, 358perf)

<https://atcoder.jp/contests/ahc050/submissions/67382558>

※満点 = 最終ターン開始時 100% 生存、ケース当たり100万点

解法2

実はちょっとした改善でスコアが伸びる

- ▶ マス (i, j) を i と j に関する2重ループで調べ、空きマスなら列 P の末尾に追加することを繰り返す
- ▶ 最後に、列 P 全体をシャッフルして出力

75,701,167 点 (満点の 50.46%, 986位相当, 649perf)

<https://atcoder.jp/contests/ahc050/submissions/67382973>

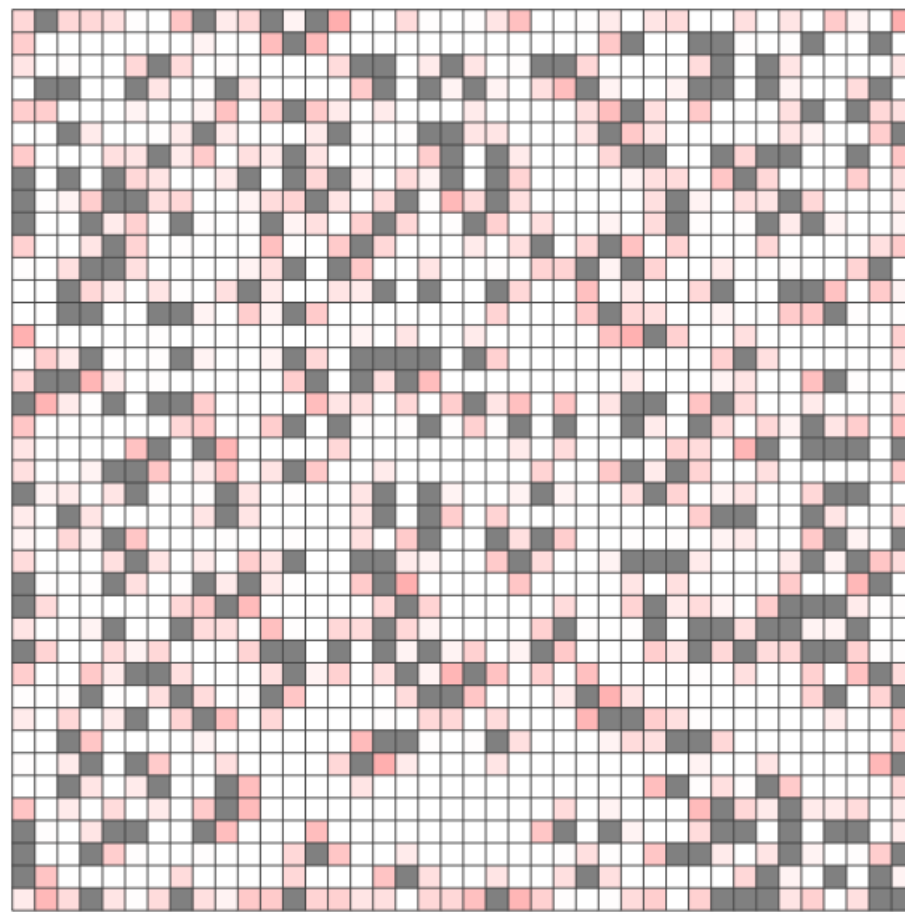
観察

ここでビジュアライザを試してみる

すると、ゲーム開始後の最初のロボットの移動で既に確率 0 のマスがたくさん存在することが分かる

そのようなマスに対して優先的に岩を置けないか？

Score = 0 (Not finished)



ジャッジを書く

ここで、この問題を Algorithm 的に解析したくなる
具体的には、以下の問題を解きたい

- ▶ 列 P が与えられるので、スコアを計算せよ

ジャッジを書く (Algorithm パート)

以下の動的計画法を考える

- ▶ $dp[\text{ターン } t][\text{マス } (i, j)] = \{t \text{ ターン目の移動が終わった時点で、ロボットがマス } (i, j) \text{ に居る確率}\}$
- ▶ (移動元マス, 移動方向) の組を決めた時に移動先がどのマスになるかを随時求めることで、この DP の遷移が行える
- ▶ 時間計算量は $O(N^5)$

ジャッジを書く (Algorithm パート)

先ほどの DP は計算量改善する

- ▶ $dp[\text{ターン } t][\text{マス } (i, j)] = \{t \text{ ターン目の移動が終わった時点で、ロボットがマス } (i, j) \text{ に居る確率}\}$
- ▶ ロボットが左へ移動する場合を考える
- ▶ このとき、各行を左から順に調べて、
 - ▶ 岩があったら移動先を次のマスに設定
 - ▶ 岩がなければそのマスから移動先へ遷移
- ▶ この方法で 4 方向の遷移すると時間計算量 $O(N^4)$ に改善

計算量？

Heuristic の問題なのに、先程から Algo の問題のように「計算量」というワードを出しまくっている

その理由は、今回の問題には特に計算量の余裕がないから

計算量？

▶ $O(N^4)$... 時間計算量は額面 2.56×10^6

▶ $O(N^5)$... 時間計算量は額面 1.02×10^8

1 秒間で処理できるのは額面 $10^8 \sim 10^9$ 程度

例えば、先ほどの解法でも DP を計算量改善しなければ試行回数は回って 10 回オーダー

$O(N^4)$ の解法すら 200 回くらいしか回らず、この回数は列の要素数 $O(N^2)$ に比べて非常に小さい

計算量を抑えて上手い解を作るよう留意しながら進む

解法3

- ▶ 各ターンごとに各マスにロボットが居る確率を求める
- ▶ 最も確率の低いマスをランダムに1つ選び、岩を置く
- ▶ ついでにスコアも計算可能

137,442,109 点 (満点の 91.62%, 565位相当, 1272perf)

<https://atcoder.jp/contests/ahc050/submissions/67404164>

解法4

- ▶ 先程のシミュレーションにはランダム性があるので、時間いっぱい回して最もスコアが良い場合を採用する
 - ▶ シミュレーションは 170 回くらい回る

141,785,929 点 (満点の 94.52%, 387位相当, 1503perf)

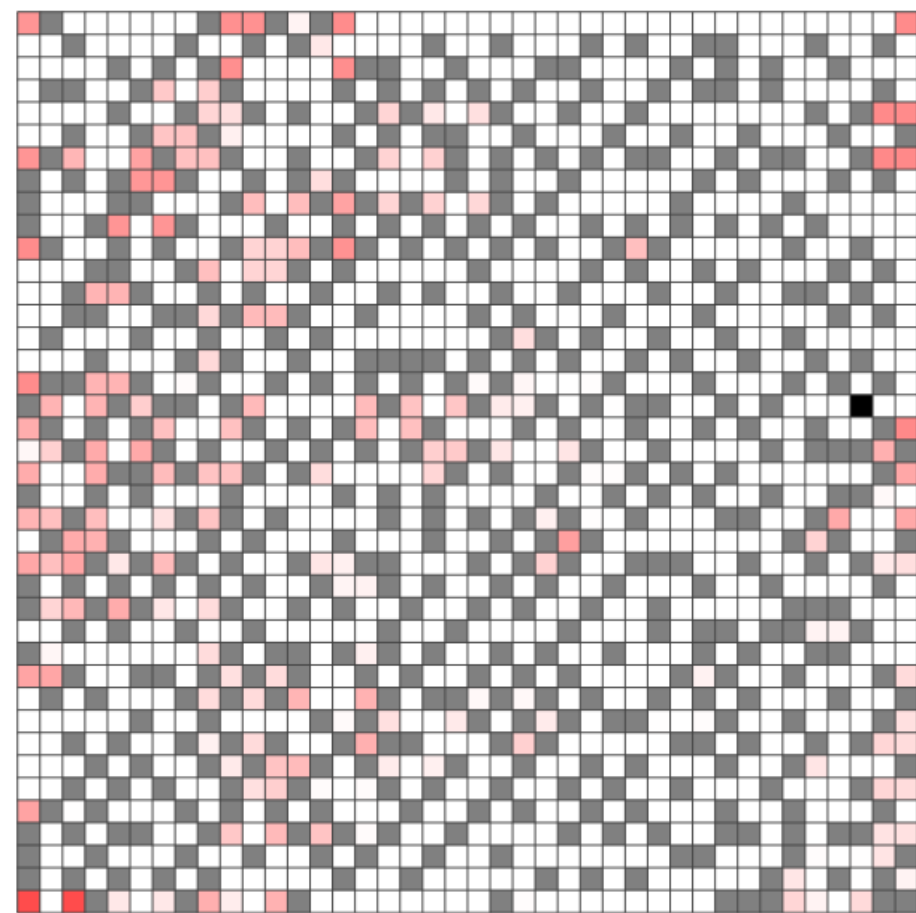
<https://atcoder.jp/contests/ahc050/submissions/67404598>

解法4の欠点と対策

ビジュアライザの途中状態を観察

ロボットを最後にまとめて潰したいのに、ロボットが存在するマスがいくつもの連結成分に引きちぎられてしまっている

特徴的な市松模様をやめさせれば連結成分が引きちぎれることは防げそう



賢いタイプレイカーの導入

先程までは、以下の値が最小になるマスを選んでいた

- ▶ (そのマスにロボットが存在する確率)

沢山ある確率 0 のマス同士の優劣がこれでは上手く比較できず、先ほどのような市松が形成されてしまう

マス (i, j) に対して $f(i, j)$ をうまく定めて

- ▶ (そのマスにロボットが存在する確率) + $f(i, j)$

と評価値を変えれば、市松を阻止できるかもしれない

解法5

ランダムに $(0, 1, \dots, N - 1)$ の順列 p, q を定めて

- ▶ (そのマスにロボットが存在する確率) $-\varepsilon \times \min(p_i, q_j)$ と評価値を変える ($\varepsilon = 10^{-6}$) と強い
- ▶ この評価値により、直線的な岩の建設が期待できる

評価値をこれに変更し、時間いっぱい回す

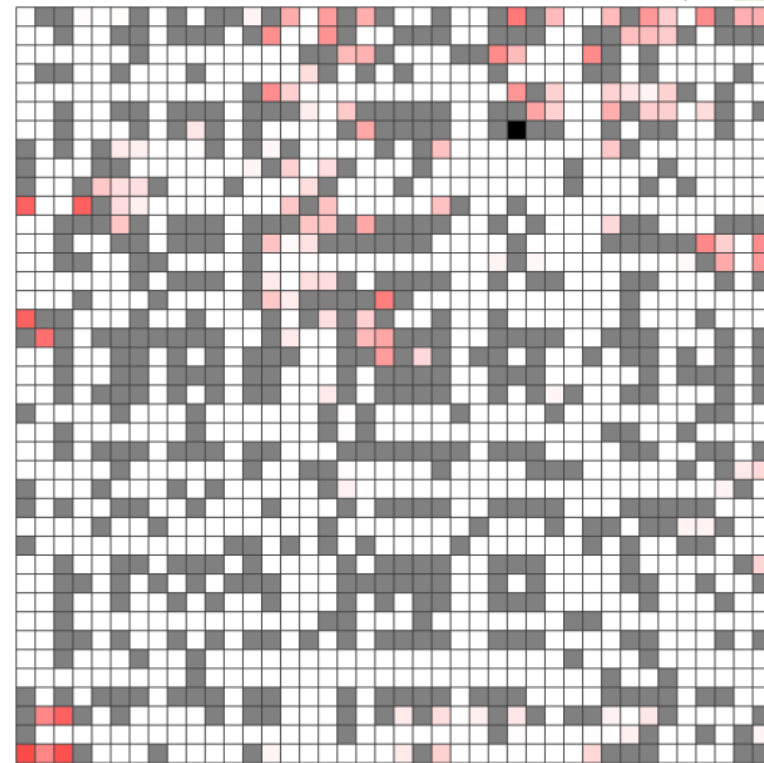
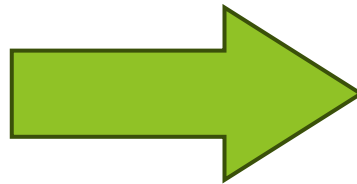
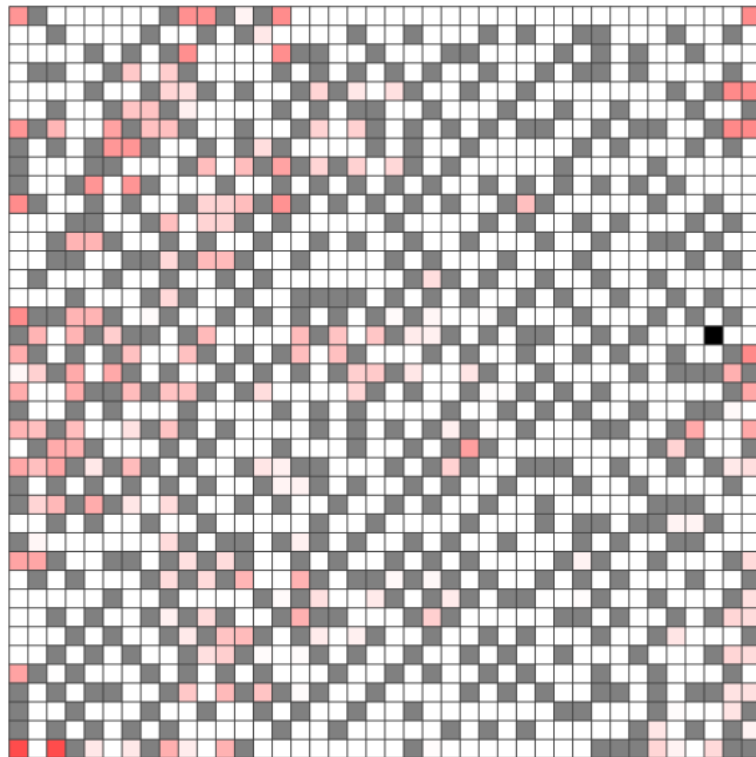
147,624,447 点 (満点の 98.41%, 82位相当, 2170perf)

<https://atcoder.jp/contests/ahc050/submissions/67405659>

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

解法5の図解

両図は250ターン目の解法4,5の比較 確かに連結成分が減った



「連結成分を減らす」の究極形

「連結成分を減らす」をモチベーションにここまで改善してきたが、これを究極まで洗練させるとどうなるか？

→ 特定の少数のマスに集中させる！

特定のマスに集中させるには?

以下の基準で各マス进行评估する

- ▶ 集中させたいマスを距離 0 とする
- ▶ 各マスについて最短何回の滑走で距離 0 のマスに辿り着くかを距離とする
- ▶ 計算量は $O(N^2)$

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

集中させたいマスはどこか？

どこに集中させる(どこを距離 0 にする)べきか？

集中させたいマスはどこか？

どこに集中させる(どこを距離 0 にする)べきか？

一例として「現在存在確率が最も高いマス群」が考えられる

- ▶ 最も存在確率の高いマスにおける存在確率を p とし、確率 $p - \varepsilon$ (ε は微小量) 以上であるマスに集めることにする
 - ▶ こうすると、複数マスで振動しているようなケースもカバーできる

計算量を抑える

毎ターン全てのマスに対して評価することはできない

- ▶ 時間計算量 $O(N^6)$ は間に合わない

以下の値の小さい方から K 個のマス进行评估する

- ▶ (そのマスにロボットが存在する確率) + (微小な乱数)
 - ▶ いくら距離が改善しようと、確率が高いマスを潰してしまってもったいない
- ▶ これで時間計算量 $O(KN^4)$ になる

解法6

- ▶ 岩を置くマスとして、以下の評価値が小さい方から $K = 10$ 個のマスを検討
 - ▶ (そのマスにロボットが存在する確率) + (微小な乱数)
 - ▶ 但し、存在確率が最小のマスに比べて 10^{-6} 以上大きければそのマスは潰さないことにする
- ▶ 岩を置いた直後の時点で距離の期待値を求め、期待値が最小となるマスに岩を置く
 - ▶ 到達不能 (距離 ∞) な場合は距離 100 とする
- ▶ 試行全体を時間いっぱい繰り返す (8,9 回動く)

149,494,453 点 (満点の 99.66%, 7位相当, **2933**perf)

<https://atcoder.jp/contests/ahc050/submissions/67441767>

解法7

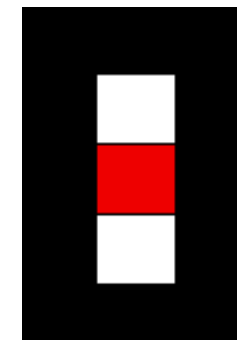
- ▶ 「距離の期待値」を「距離を適切に写像したものの期待値」と取り換える
 - ▶ 例えば $f(x) = x^2, f(x) = \exp(x), f(x) = \log(x), \dots$
 - ▶ 各距離を写像したものの配列を定義しておくと、中身を簡単に変更できて便利
- ▶ 今回は $f(x) = \min(4^x - 1, 10^{18})$ を採用
 - ▶ 写像後の値が大きくなるので、開始時に最大でない連結成分に閉じ込められているケースは評価値に算入しないようにする

149,603,218 点 (満点の 99.73%, 4位相当, 3085perf)

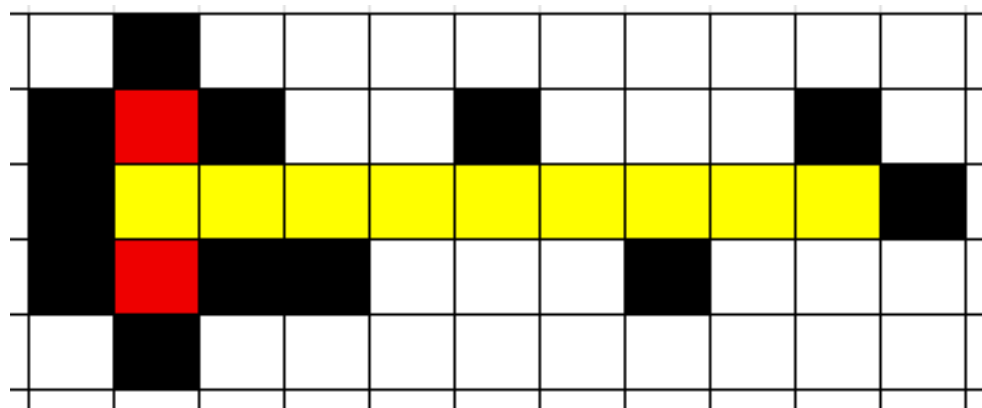
<https://atcoder.jp/contests/ahc050/submissions/67443150>

理想の終局を作りに行く

理想的には、最後 3 手が右の状態になっていればよい

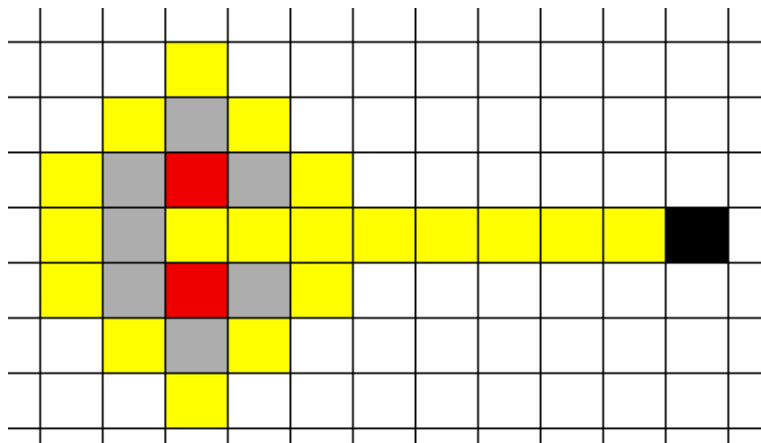


これを実現するためには、下図の途中状態を経由しそう



理想の終局を作りに行く

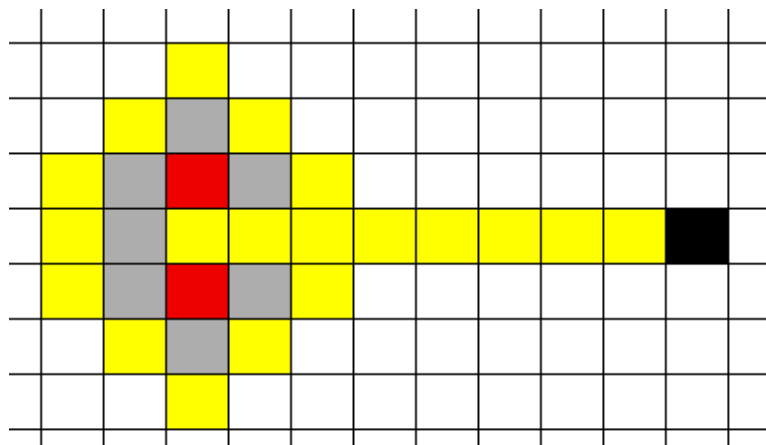
- ▶ グレー ... 岩の有無は問わないが、早い段階で岩を置く
 - ▶ 但し、グレーに岩を置いて最大の連結成分が分割されてはいけない
- ▶ 黄色 ... 岩があると邪魔になるので、岩がないべき
- ▶ 赤色 ... 最後に潰す 2 マス、この 2 マスを距離 0 とする
- ▶ この中で、横伸びがなるべく長い部分グリッドを取る



理想の終局を作りに行く

道中では以下の処理を実装

- ▶ グレーマス ... 存在確率が十分小さければ、直ちに潰す
- ▶ 黄マス ... そのマスを埋める優先順位を下げるため、存在確率に下駄を履かせて潰されにくいようにする

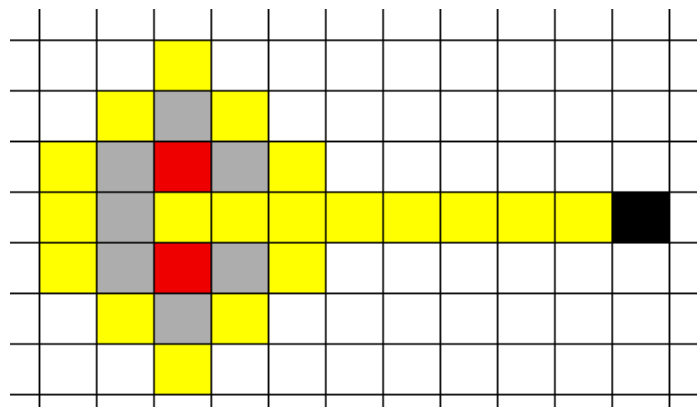


本当に実現可能か?

本当にこのような部分グリッドが存在するかが懸念

ワーストの $1/4$ が最初から岩だとしても、場所を固定した時に必要最低限の空白 (16マス) は $(3/4)^{16} = 1\%$ 程度で取れる

→ グリッド全体を見た時、高い確率でどこかには存在しそう



小手先改良

- ▶ 求めた列を P とする
- ▶ P を先頭から見て、各マスごとの存在確率を求める
- ▶ P を末尾から見て、各マスごとの残り生存ターン数の期待値を求める

これを利用すると次のような改良が 1 回あたり $O(N^4)$ で可能

- ▶ 1 項掴んで insert のうち最適なものを実行
- ▶ 隣接 swap のうち最適なものを実行
- ▶ これらの改良を 1 秒あたり 100 回程度回せる

解法8

- ▶ 解法7に先程の「理想の終局」のルールベースを加える
- ▶ 低いスコアしか出ない場合があるので、その場合は解法7かつ「真ん中あたりのマスに集める」とした解法を回す
- ▶ これらで作られた解のうち最も良いものを暫定解とし、「要素を1つ掴んで最適 insert」を末尾から始めて時間が来るまで繰り返す
- ▶ これらを組み合わせたうえで、細部を頑張って調整する

149,892,531 点 (満点の **99.92%**, **1位相当**, **3534perf**)

<https://atcoder.jp/contests/ahc050/submissions/67448254>

余談

- ▶ 今回の AHC050 、 0点の順位が 1086 位
- ▶ つまり、正の得点を取った方は 1085 名
- ▶ これは Rated 短期 AHC の中では史上最大っぽい

沢山のご参加ありがとうございました！