

# ICPC模擬国内予選2017

## H: Big Maze

原案: nodchip

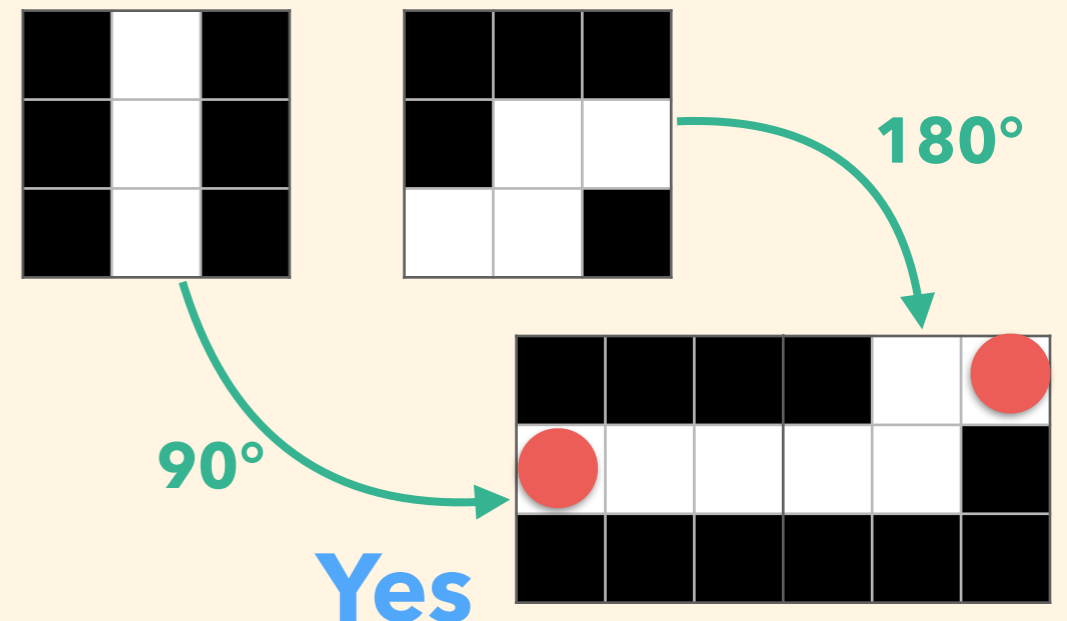
問題文: kurome

解答: darsein, kurome

解説: darsein

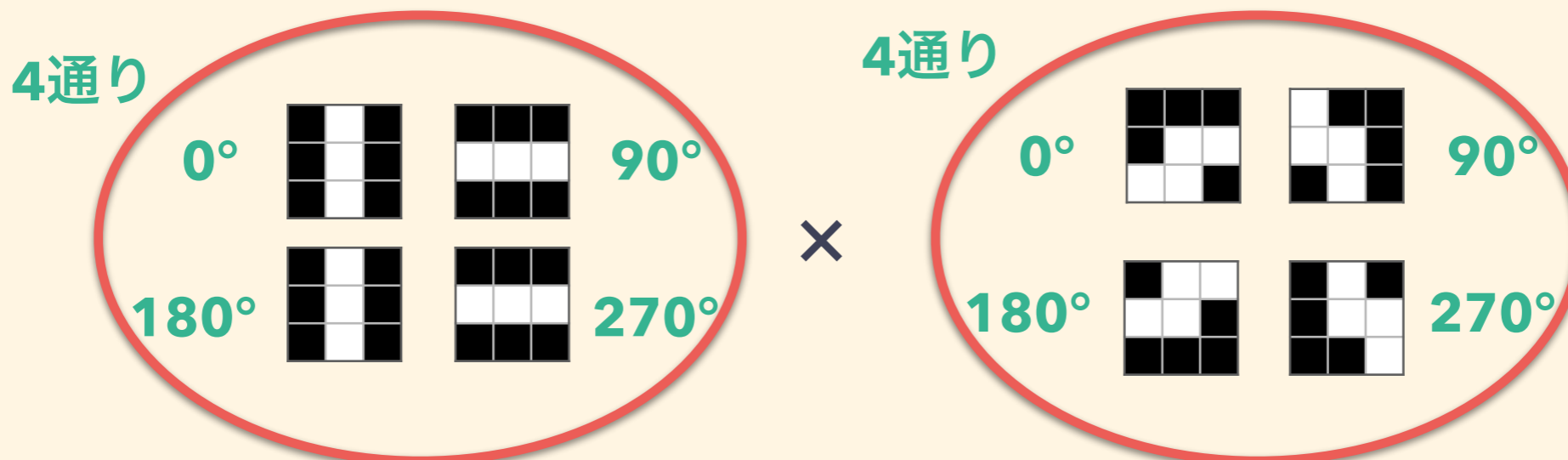
# 問題概要

- $N \times N$  サイズのブロック  $M$  個を1列に並べて迷路を作りたい
  - ブロックの各マスは壁か通路
  - 通路のみ通れ、移動は上下左右の4方向
- 並べる時は各ブロックを順番通り使う必要があるが、 $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ 回転させてもよい
- 左端の通路マスから右端の通路マスに移動可能なように迷路を作れるか Yes/No を答えよ
- 制約:  $1 \leq N \leq 12$ ,  $1 \leq M \leq 1,000$



# 想定 TLE 解法: 全探索

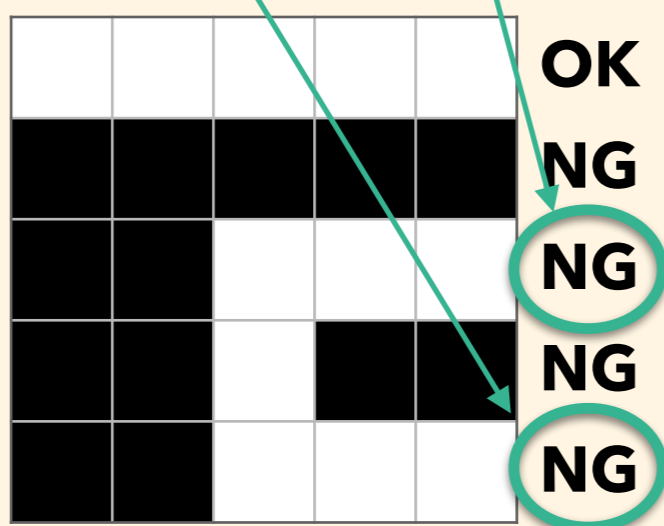
- 回転角度を決めうちすれば、あとは接続してできた迷路がゴール可能か判定するだけ
  - BFS等の探索アルゴリズムで線形時間
- 回転角度の組合せが $4^m$ 通り、探索が $O(N^2M)$ 
  - 全体で  $O(N^2M 4^m)$ ,  $m=1000$ なのでTLE



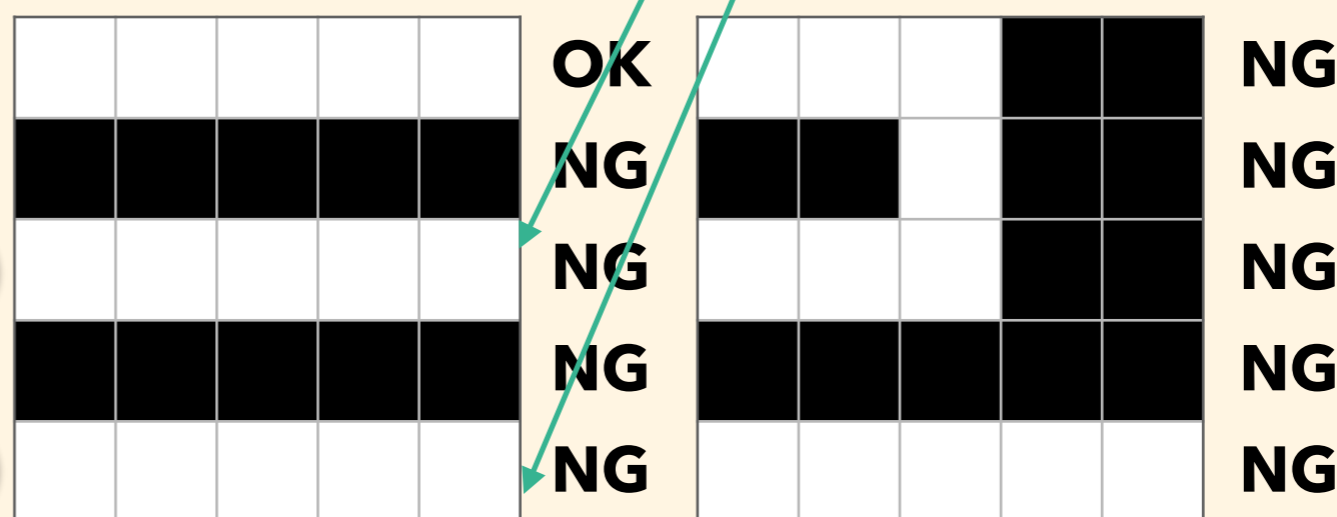
# よくある間違い

- 左端から到達可能なマス集合を覚える動的計画法  
(プロコン界隈でのいわゆる bit DP)
- 折り返しがある時に死ぬ (サンプルに入っている)

ここまでの情報だと左端から  
行けないので使えないと判断

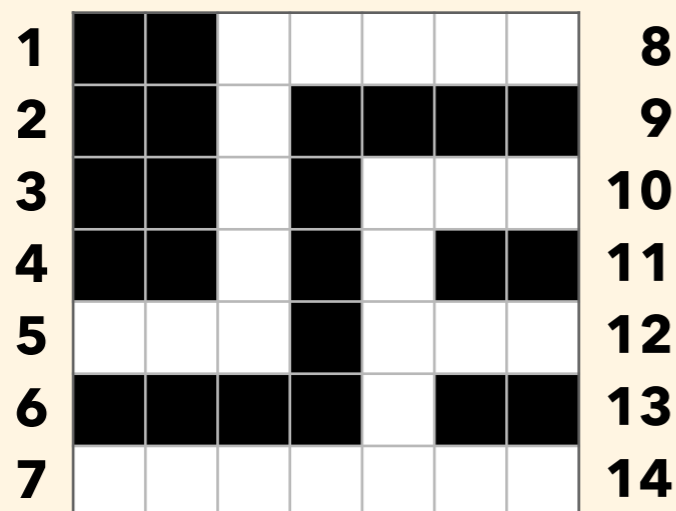


実際は後々折り返してきて  
使える (こともある)



# 想定解法: (めんどくさい) DP

- 左から1つずつブロックを足していく
- 現状右端の各マスと左端のマスの“到達可能性関係”を記憶する (例: 3,4,7番目のマスは相互到達可能など)



**5,8 が相互到達可能**

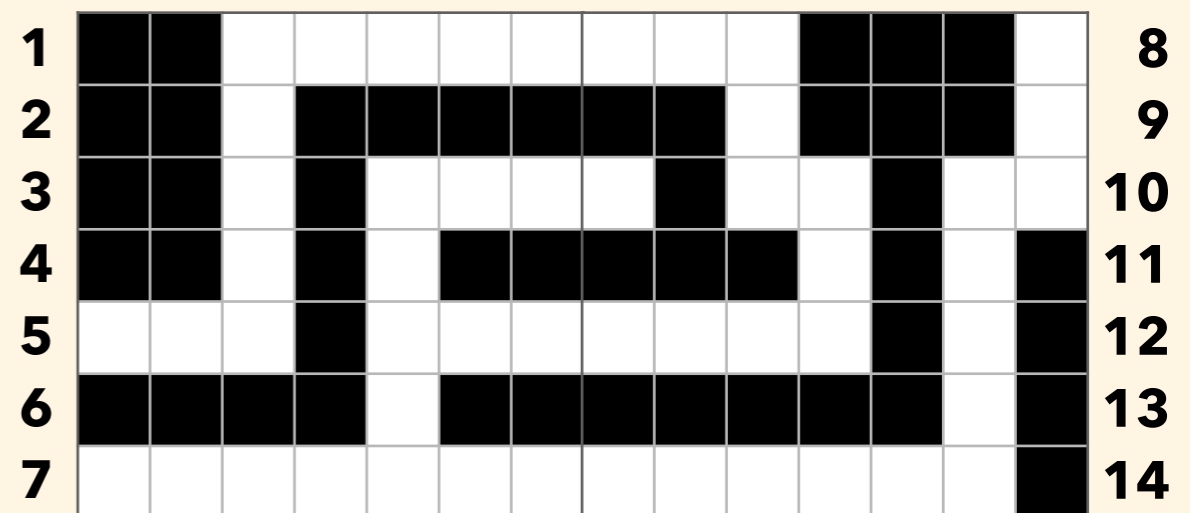
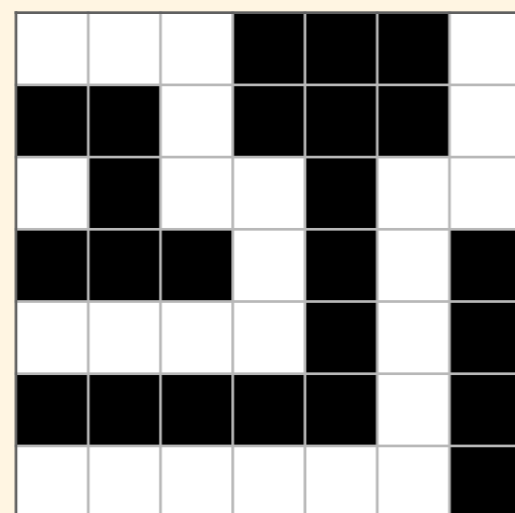
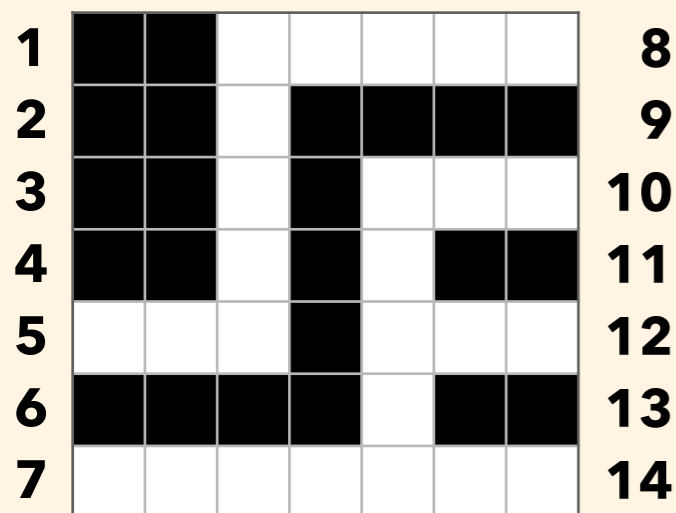
**7,10,12,14 が相互到達可能**

# 想定解法: (めんどくさい) DP

- 新たに追加するブロックの左端・右端の到達可能性と、現状の到達可能性から、新たな到達可能性を計算

**5,7,8,9,10 が相互到達可能**

**(5,7は実際区別しなくてよい)**



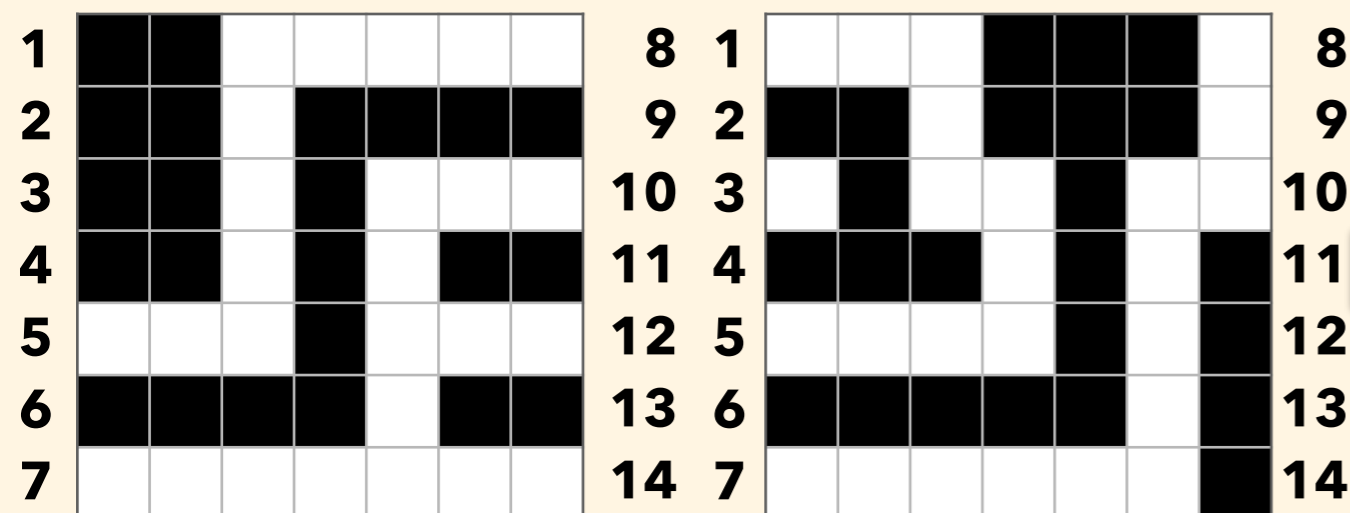
# 想定解法のDP表

- $dp[i]$  [ブロック目][右端Nマス+左端まとめて1マスの到達可能性を表す表現] := Yes / No
- 到達可能性は集合分割の形で書ける
  - 1, 8 が相互到達可能、かつ 3, 4, 7 が相互到達可能 →  $\{\{1,8\}, \{3,4,7\}\}$
  - 連想配列で集合分割をキーにすると実装が楽
    - No のケースはそもそも記憶しない
    - 高速・省メモリ化

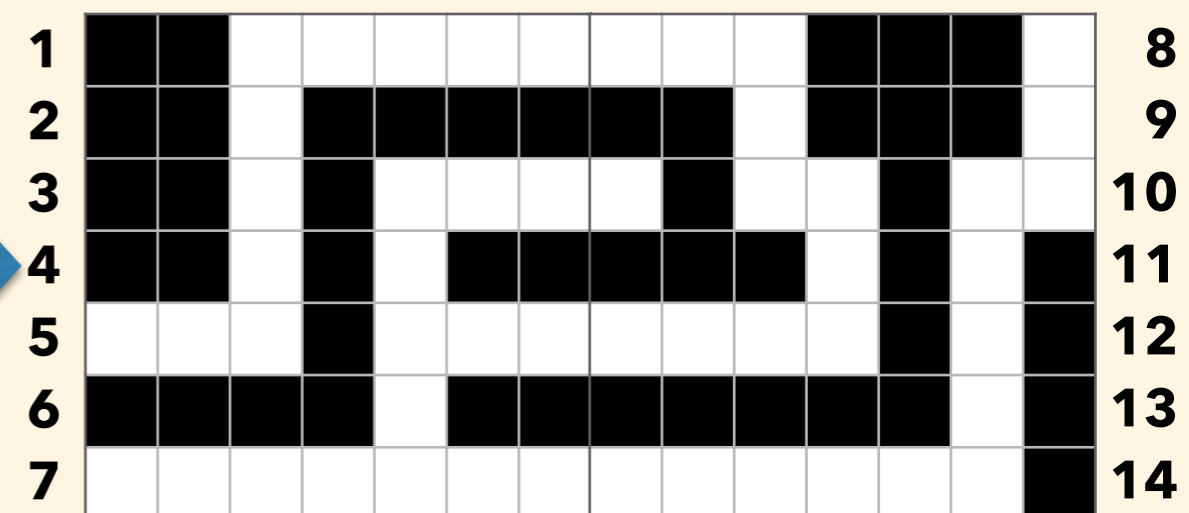
# 想定解法のDP更新

- あらかじめブロックの回転ごとに左端・右端の到達可能性を前計算しておくことで計算量が落ちる
- Union-Find 木を使うことで線形時間 ( $O(N)$ ) で更新が可能

**{{5,8}, {7,10,12,14}}**    **{{1,5}, {3}, {7,8,9,10}}**



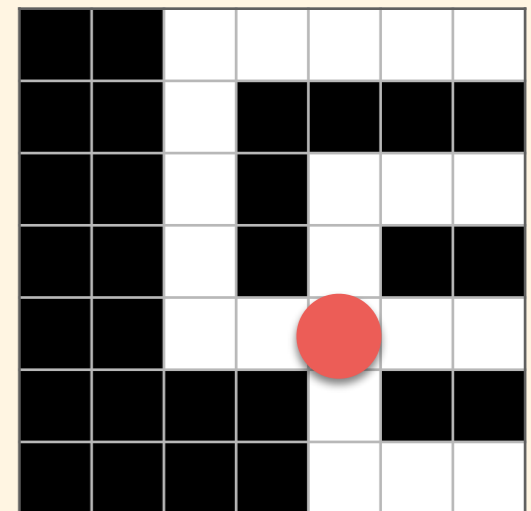
**{{5,7,8,9,10}}**





# 想定解法の計算量

- Union-Find の計算量を  $O(\alpha(N))$  とすると、
  - 各ブロックの到達可能性の前計算:  $O(N^2M)$
  - DPの計算:  $O(M \cdot |\text{Yesの集合分割の個数}| \cdot N \cdot \alpha(N))$
  - $|\text{集合分割の個数}|$  はベル数  $B_n$  で抑えられる
    - $B_{12} = 4,213,597$  と大きいのが、
      - 隣り合う通路は別の集合に入らない
      - 交差しない
    - などの制約から実際はずっと少ない



# 余談

- このような DP はフロンティア法と呼ばれる
  - c.f. <http://www-lsm.naist.jp/~jkawahara/frontier/>
  - 実際のフロンティア法はDPテーブルの代わりにZDDと呼ばれるデータ構造を作る
- 国内予選で TLE 3時間なので雑でも間に合うが、高速化しようとする実装が結構大変

# Writer 解

- darsein: 205 行 5011 bytes (C++)
- kurome: 194 行 4727 bytes (C++)

# 統計情報

- AC / trying teams
  - 1 / 5 (20.00%)
- First Acceptance
  - 非現役込み: TEAM NAGATO (186:56 (延長込))
  - 現役のみ: TEAM NAGATO (186:56 (延長込))