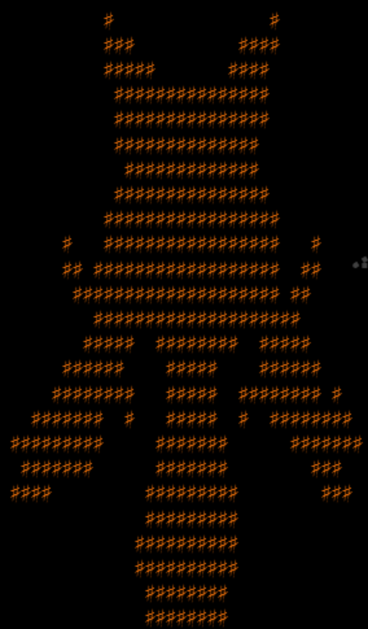
Soul Gem Game

-JAG 2011 summer camp day 4, Problem J-

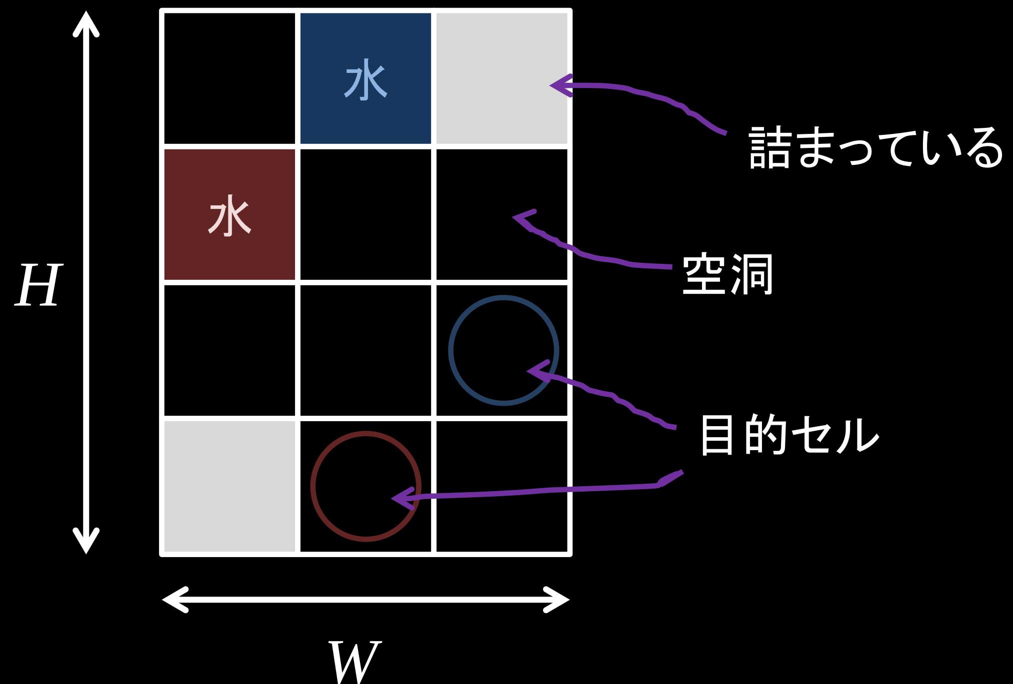


writer : ir5

tester : cos

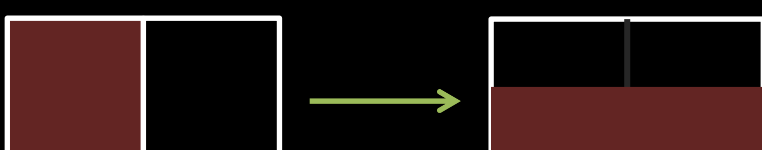
概要

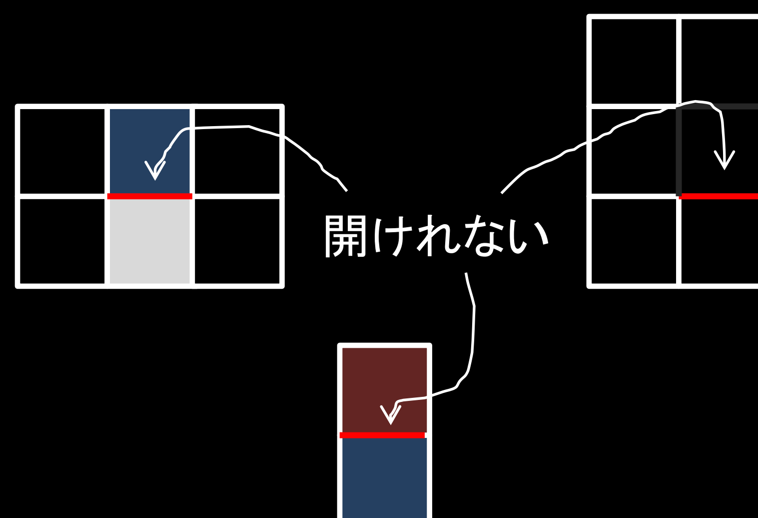
- $H \times W$ の水槽がある
- 青・赤の水がそれぞれ1つのセルに入っている
- 青・赤の水それぞれについて移動させたい目的地のセルが指定されている
- 他のセルは空洞か，中身が詰まっている(水を入れれない)



概要

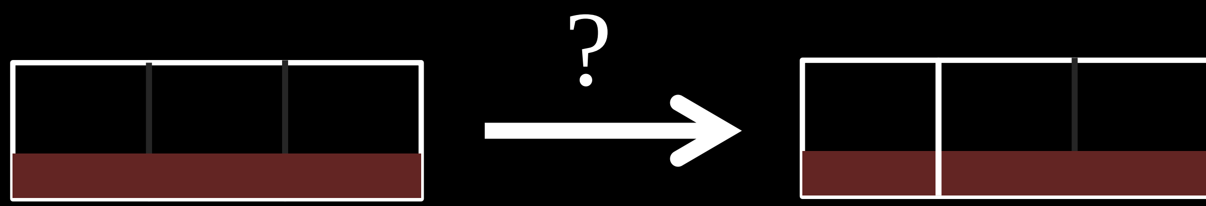
- コスト1 で隣り合う水槽間の壁を開閉できる
- 水槽の壁を開くのには以下の制約がある
  - 1つのセルの隣の壁は同時に3つ以上開いてはいけない
- 水は物理法則に沿って動く
- 青と赤の水を混ぜてはならない
- 水を目的セルまで移動させたい．最小コストを求めよ
- 制約 : $H, W \leq 10$



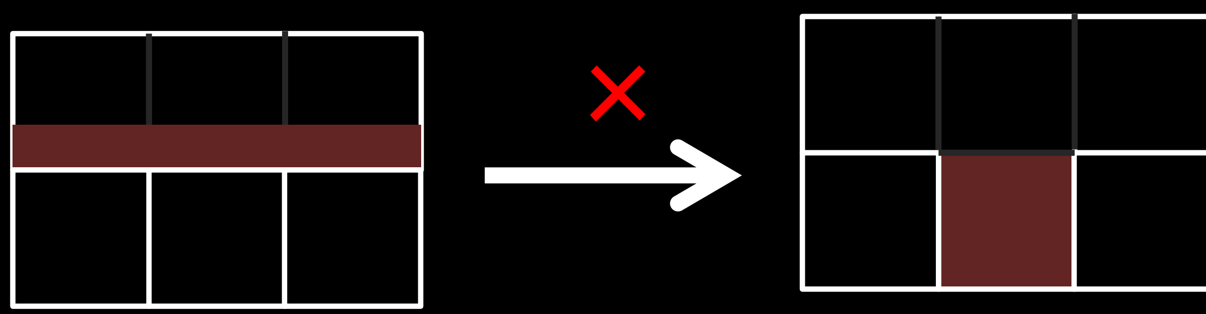


簡単な観察

- 水が横に伸びてるときに水の間の壁を閉じるのは意味無い
  - 分裂させた分コストが増えるだけ

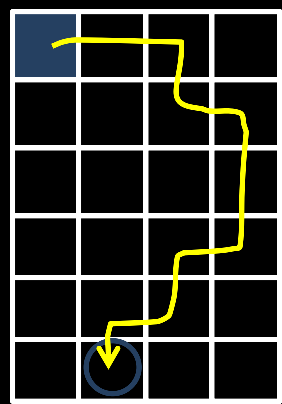


- 水が通った跡をT字型みたいにするのはできない



水路・停留セル

- 先程の考察より，水が通った跡はパスとして図示できる
  - ↓の P のセルを水路と呼ぶことにする



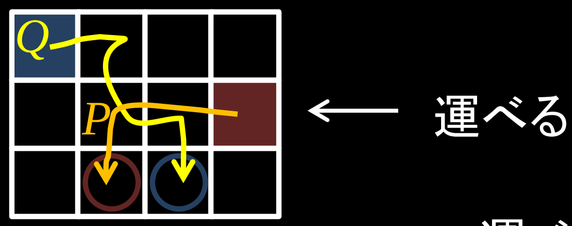
```
PPP.
..PP
...P
..PP
.PP.
.P..
```

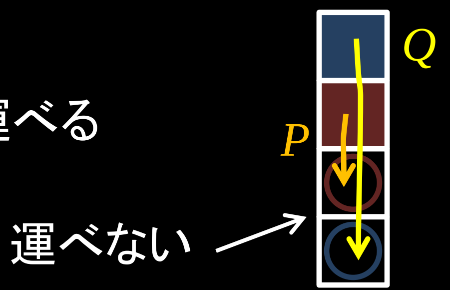
- 水路に含まれるセルで，↓の @ のようなセルを停留セルと呼ぶ

```
@PP.
..@P
...@
..P@
.P@.
.@..
```

水路 P, Q で水を運べる条件

- 2本の水路 P, Q を取った時，実際にその水路を通って水を運べるか，また，コストはいくらになるかについて考える



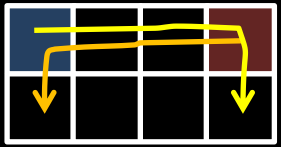


- 場合分けで以下が証明できる.
  - 水路の組 P, Q を用いて2つの水を運べる

$\Leftrightarrow$ [Q の停留セルで P に含まれていないものがある]
and [P の停留セルで Q に含まれていないものがある]
and [[Q の始点が P に含まれていない]
or [P の始点が Q に含まれていない]]

水路 P, Q で水を運べる条件

- [⇒]
  - 条件のどれかを満たさないときを考える
  - [Q の停留セルはどれも P に含まれる] とき: どうやっても水が混ざるので無理
  - [Q の始点が P に含まれる] and [P の始点が Q に含まれる] とき : 衝突せざるをえないので水が混ざって無理



- [⇐]
  - 3番目の条件より, いきなり衝突することはない
  - P の終点の停留セルは Q に含まれてないとする. このとき Q の水を, P に含まれない停留セルまで移動させる→ P の水を終点まで動かす → Q の水を終点まで動かす という風にすればよい

水路 P, Q のコスト

- 問題は次のように書き換えられる.
  - 水路 P, Q の始点，終点のセルがそれぞれ与えられるので，水を運べる水路 P, Q でコストの最小を求めよ.

- 水路のコストについて考えたい.

- 基本的には コスト $= |P| + |Q|$
- パスが共通している場合，そこは使いまわせるのでコストを省ける
- 一方の水路がもう一方に``侵入”するとき，壁を閉めないといけないので 1 のコストが発生する

水路 P, Q のコスト

- P を固定して，コストが最小になる Q を求めるとする
- 最小コストはダイクストラ法で計算できる

```
PPP..Q
..PP..
###P##
##PP##
.PP...
QP....
```

- ↑で上の Q から下の Q に移動するとして，
  - [.] ↔ [.] のコストを 1
  - [.] ↔ [P] のコストを 2
  - [P] ↔ [P] のコストを 0
  - として最短パスを計算する(始点，終点に例外あり)

水路 P, Q のコスト

- しかし実際には P, Q で水を運べない場合を弾く必要がある

- Q は P の始点を含まない(P は Q より上にある) として良い
  - これより，気にすべきことは [Q の停留セルで P に含まれていないものがある] だけになる

- 上の Q を始点にダイクストラ，下の Q を始点に逆向きにダイクストラし，Q の停留セルで P に含まれない点を停留セルにするようなものを全部列挙して答えを求める

- 全体で $O(HW \log(HW))$ くらいで計算できる

水路 P の列挙

- 水路 P を列挙することを考える
- 超単純にやると $O((H\text{-}1)^W)$ 個列挙とかでアカン
- $O((H\text{-}1)^W)$ 個の中にはクネクネしてるケースが多い
  - が，クネクネしてるのは最適解からは遠い
  - ので，枝狩りする

```
PPP......
..P......
#.PPP....
##..PP...
###..P.##
.....P...
..PPPP.##
..PPP...#
##..PPP..
###...P..
```

クネクネしてる

水路 P の列挙

- 即座にUターンしてるようなのは無意味

```
##.PPP.        ##.PPP.
##PP###   =>   ##.P###
..PPP.#        ...PP.#
....P.#        ....P.#
```

- ``コ”の字型のところも省ける
  - ``コ” の空洞には # や Q の始点，終点が無いとする

```
P...Q.        P...Q.
PPP.##        PPP.##
..PPP.        ..PP..
.##.P.   =>   .##P..
#...P.        #..P..
.#PPP.        .#PP..
..P..Q        ..P..Q
```

``コ”の字を枝狩りできる理由

- 枝狩りする前の P に対して, Q を最適解を達成する水路とする.

- ``コ”を枝狩りすると Q の停留セルが P に含まれてしまうとき
  - Q も変えることによって解を改善できる
  - P 側:-2 ; Q 側:高々+1

```
P.....    ......
PPP.##    ..Q.##
..PPP.    ..QQ..
.##.P.    .##Q..
#...P.    #..Q..
.#PPP.    .#.Q..
..P...    ......
```

```
P.....    ......
PPP.##    ..Q.##
..PP..    ..QQQ.
.##P..    .##QQ.
#..P..    #..Q..
.#PP..    .#.Q..
..P...    ......
```

``コ''の字を枝狩りできる理由

- それ以外のとき
  - ``コ''を枝狩りすることによって増える Q 側のコストは高々 2 である
  - ``コ''を枝狩りすることで P 側では 2 のコストが減るので, 解が悪化することはない

- 枝狩りのオーダーはよくわからないけど速い(ので間に合う)

結果

- 2011年度 ICPC 夏合宿最終日に使用
  - 0 Accepted
- Aizu Online Judge に収録されている
  - 2012年4月25日現在，ジャッジしか Accepted してない……

余談

- SRM507のDiv1::Hardとして問題を書きました(最初は $H \leq 8$, $W \leq 8$だったので解が十分速いことは保証していた)
- 使われなかったので JAG で使いました

- 状態をうまくおけば多項式時間で解けるのではないかと予想していた気がするのですがよくわかってません