

ACM-ICPC JAG

模擬国内予選 2015

C: ブロック落とし

原案：森

問題文：水野

テストケース：山崎

解答：澤・保坂・水野・森・山崎

解説：山崎

問題概要

- 三次元落ち物パズルゲームをする
 - 盤面の全体は, 横幅 2 奥行き 2 高さ無限大の三次元グリッド
 - 同じ高さにある 4 つのセルが全てブロックで埋まると消える
 - 段が消えたとき, それより上にあるブロックは丁度 1 段ずつ下に落ちる
 - 初期状態で, ブロックが $2 \times 2 \times H$ ($H \leq 10$) の範囲内にいくつかある
- $2 \times 2 \times 2$ の範囲に収まるブロック塊が落ちてくる
 - 落ちてくるブロック塊の数は N ($N \leq 3$) 個
 - 落とす前に盤面からはみ出ない範囲で水平に動かすことができる
 - いずれかのブロックが床か他のブロックに接すると全体が止まる
- 初期盤面と落ちてくるブロック塊の情報が与えられる
- 最大で何段消せるか?

考察

- 水平方向に 2 セル分動かすと確実に盤外に出てしまう
- それぞれの軸(前後・左右)について, $-1, 0, +1$ しかできない
- 従って水平移動の仕方は高々 9 通り
- ブロック塊全部の落とし方は高々 9^N (≤ 729) 通り

解法

- ブロック塊の落とし方を全て試す
- 落下と消滅をシミュレート
- それぞれ消える段数が分かるので、最大のものを出力

深さ優先探索 (DFS : Depth First Search)

- 全ての解を列挙したいときに便利
- 再帰関数で実装すると楽
 - n 個処理した状態から消せる段数の最大値を返す関数を作る

擬似コード

```
int dfs( int n )  
    if ( n == N ) return 0  
    res = 0  
    foreach i 番目のブロック塊の, 水平移動の全パターン  
        if ( 不正な水平移動 ) continue  
        ブロックを落下させる  
        埋まった段を消す  
        res = max( res, 消えた段数 + dfs( n + 1 ) )  
    return res
```

水平移動

- 前後方向に dy , 左右方向に dx 動かすとする
- 位置 (i, j, k) のブロックは $(i, j + dy, k + dx)$ に移動
- ブロックが盤外に出ない $\Leftrightarrow 0 \leq j + dy, k + dx < 2$
- 全ブロックが条件を満たせば妥当な水平移動

落下

- 落下が止まる位置を求めたい
- 仮に, 落下が止まる位置の 1 段下まで下ろしたとする
- このとき, ブロックが盤面上のブロックまたは床と重なる
- 落下が止まる位置は, 初めて重なった段の 1 段上
- 重なるまで下ろしてから 1 段持ち上げれば求まる
 - 初期位置は適当に十分高い位置

消滅

- 埋まっている段を消していく
 - ついでに消えた段数を数える
- 埋まっているかのチェックは単純なループで実現できる
- 段が埋まっていたら, そこより上全部を 1 段下げる
 - 盤面を `std::vector` で保持しておけば `std::vector::erase` を利用できる
 - 線形時間かかるが, ブロックが存在する範囲は小さいので問題無し
- 連続する 2 段が消える場合にちょっと注意
 - 下から処理すると, 同じ高さで 2 回消えることになる
 - 段が消えたらループ変数が進まないようにする

まとめ

- やるべきことは単純な全探索
- ただし実装は重め
- かなりのチームがここで詰まっていました

- いかにかバグらせずに書けるか
 - 例えばインデックスのミスが発生しやすいので注意
- ペアプログラミングなどでミスに気付きやすくしましょう

ジャッジ解

- 澤 : 83 Lines / 2,553 Bytes (C++)
- 保坂 : 101 Lines / 2,942 Bytes (Java)
- 水野 : 164 Lines / 3,894 Bytes (C++)
- 森 : 177 Lines / 3,898 Bytes (C++)
- 山崎 : 194 Lines / 3,303 Bytes (C++)

結果

- First Acceptance
 - Mutability (00:37:56)
- AC / Submissions
 - 39 / 60 (65%)
- AC / Trying Teams
 - 39 / 43 (91%)