

C: Dowsing Machine

- ダウジングマシーン -

原案：井上・青木

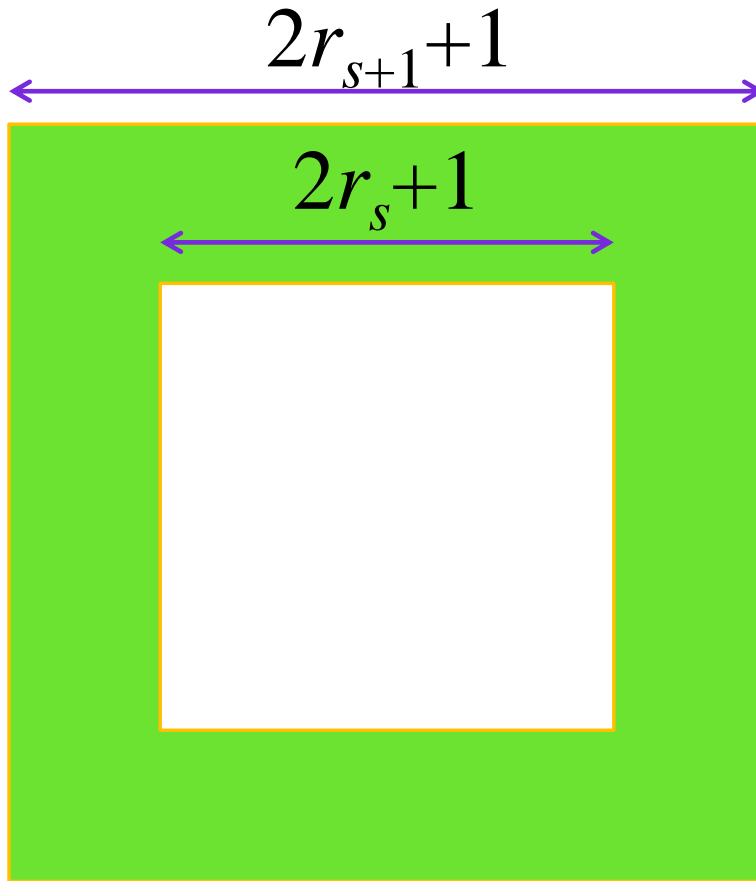
解答：井上・青木

問題概要

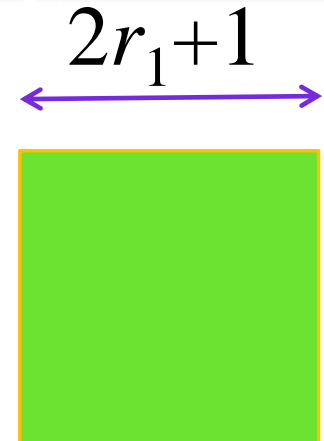
- 二次元格子のマス上でのダウジング情報が与えられる
- 財宝があるマスに到達できるか調べよ
 - Yes: 到達可能
 - No: 到達不可能
 - Unknown: 不明
 - Broken: 矛盾

Dマシンの示す反応

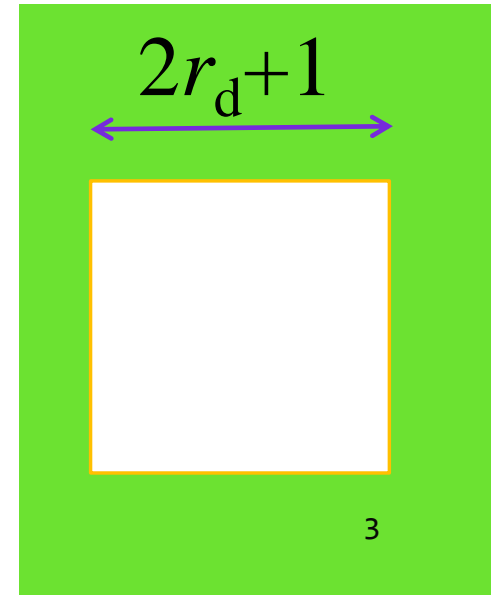
$1 \leq s \leq d-1$ のとき



$s = 0$ のとき



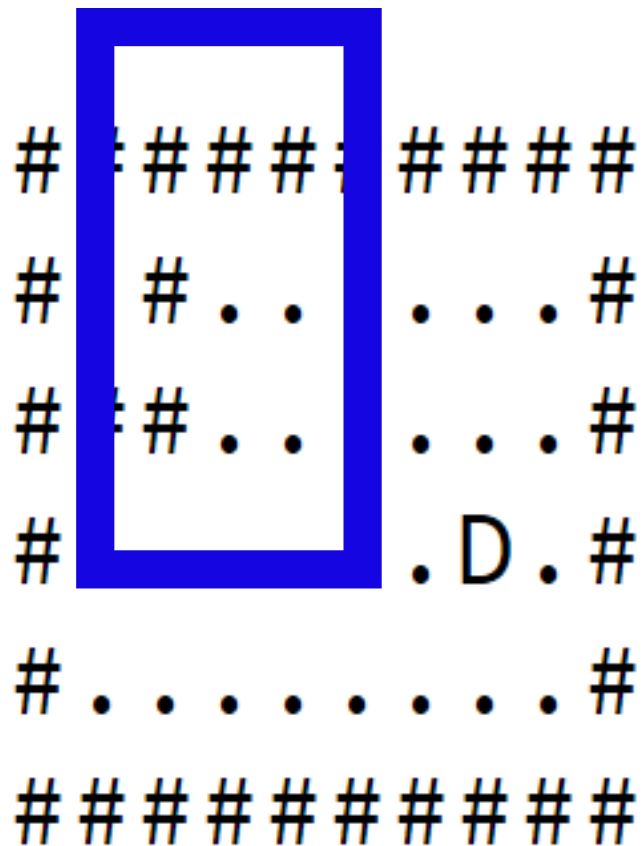
$s = d$ のとき



例

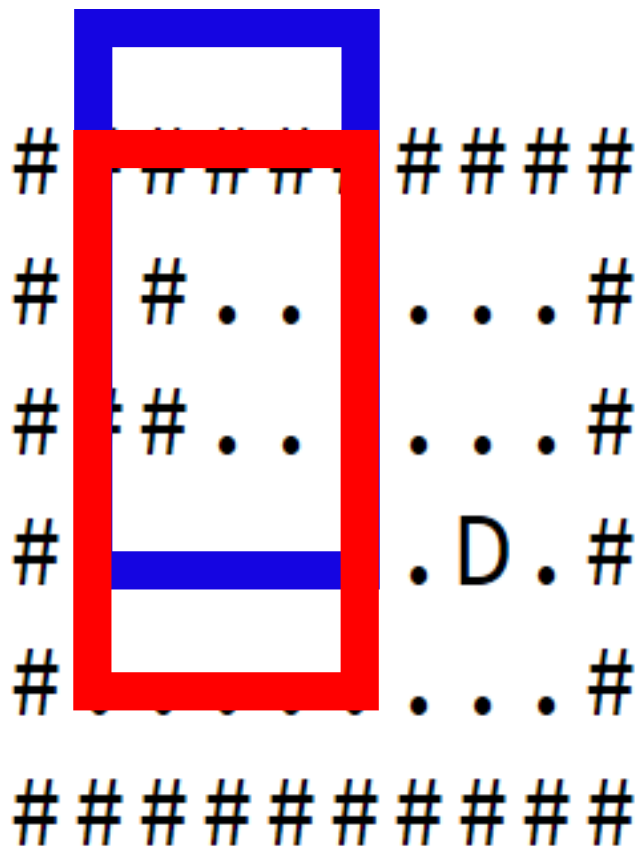
```
#####  
#.#.....#  
###.....#  
#.....D.#  
#.....#  
#####
```


例



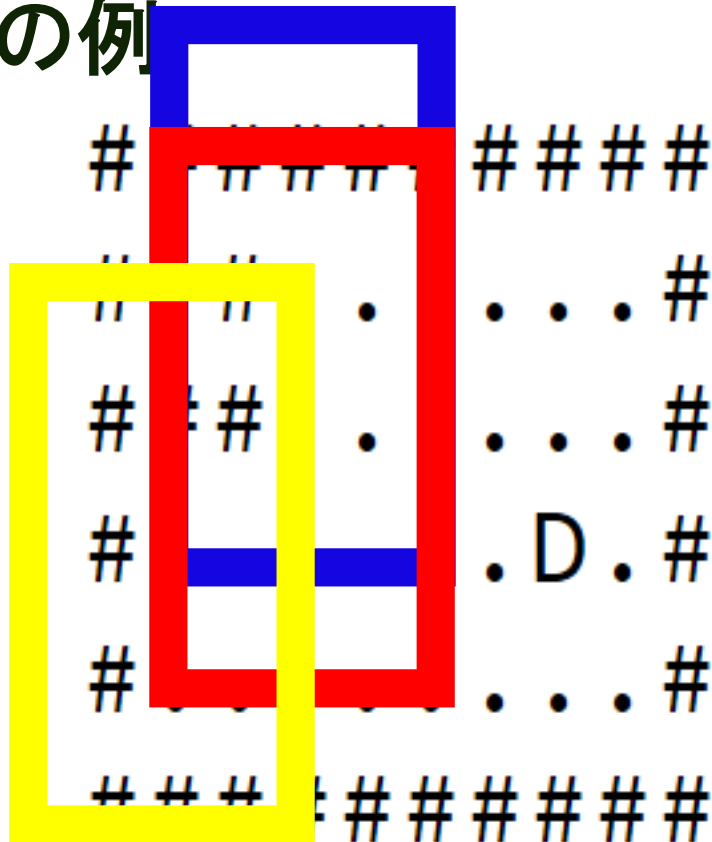
```
# # # # # # # # # #  
# # . . . . . #  
# # . . . . . #  
# . D . #  
# . . . . . #  
# # # # # # # # # #
```

例

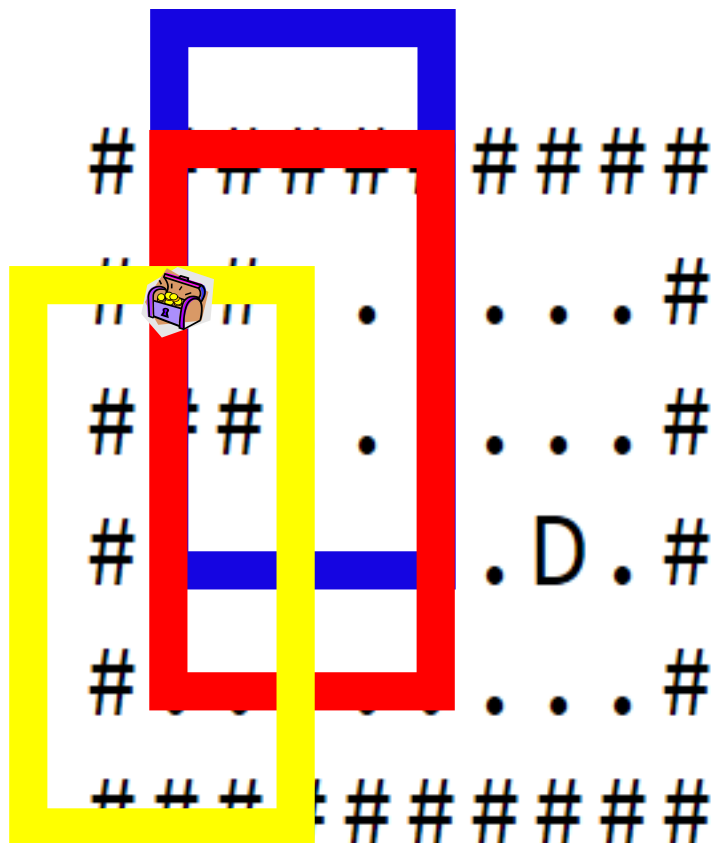


例


□ 4つの例



例

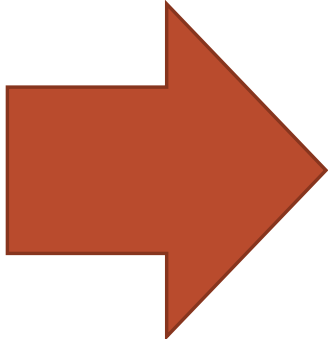


例

```
#####  
## . . . . . #  
### . . . . . #  
# . . . . . D . #  
# . . . . . . . #  
#####
```

例

```
#####  
# 🗡️# .....#  
### .....#  
# .....D.#  
# .....#  
#####
```

 No

想定解法(1/3)

- 財宝がないマスをもっと取り除き、残ったマスが財宝がある場所の候補
- 到達可能な財宝マスの数と到達不可能な財宝マスの数で4通に場合分け
- 到達判定はDFSやBFS

想定解法(2/3)

財宝がないマスをもっと取り除き、残ったマスが財宝がある場所の候補

```
u[i][j] = { true, true, ..., true };
for (int k=0; k<n; k++) {
    cin >> x >> y >> s;
    for (int i=0; i<h; i++) for (int j=0; j<w; j++) {
        if (s > 0) { // s-1の内側に財宝がない
            lx = x-r[s-1]; rx = x+r[s-1];
            by = y-r[s-1]; ty = y+r[s-1];
            if (lx <= j && j <= rx && by <= i && i <= ty)
                u[i][j] = false;
        }
        if (s < d) { // sの外側に財宝がない
            lx = x-r[s]; rx = x+r[s];
            by = y-r[s]; ty = y+r[s];
            if (!(lx <= j && j <= rx && dy <= i && i <= uy))
                u[i][j] = false;
        }
    }
}
```

想定解法(3/3)

到達可能な財宝マスの数と到達不可能な財宝マスの数で場合分け

```
reachable = noroute = false;
for (int i=0; i<h; i++) for (int j=0; j<w; j++)
    if (u[i][j])
        if (bfs(j, i))    // マス(j, i)からDの位置までの経路があるか
            reachable = true;
        else
            noroute = true;
if (reachable && noroute)    // 一部に到達可能で、一部に到達不可能
    cout << "Unknown" << endl;
if (reachable && !noroute) // 全てに到達可能
    cout << "Yes" << endl;
if (!reachable && noroute) // 全てに到達不可能
    cout << "No" << endl;
if (!reachable && !noroute) // 財宝マスの候補がない
    cout << "Broken" << endl;
```


ジャッジ解

- 井上1 (C++)

- 84行

- 井上2 (C++)

- 101行

- 青木 (Java)

- 86行

結果

□ Accept / Submit

□ 12 / 52 (23.08%)

□ First Accept

□ Online: takapt0226さん (48分)

□ Onsite: チーム yz (56分)