

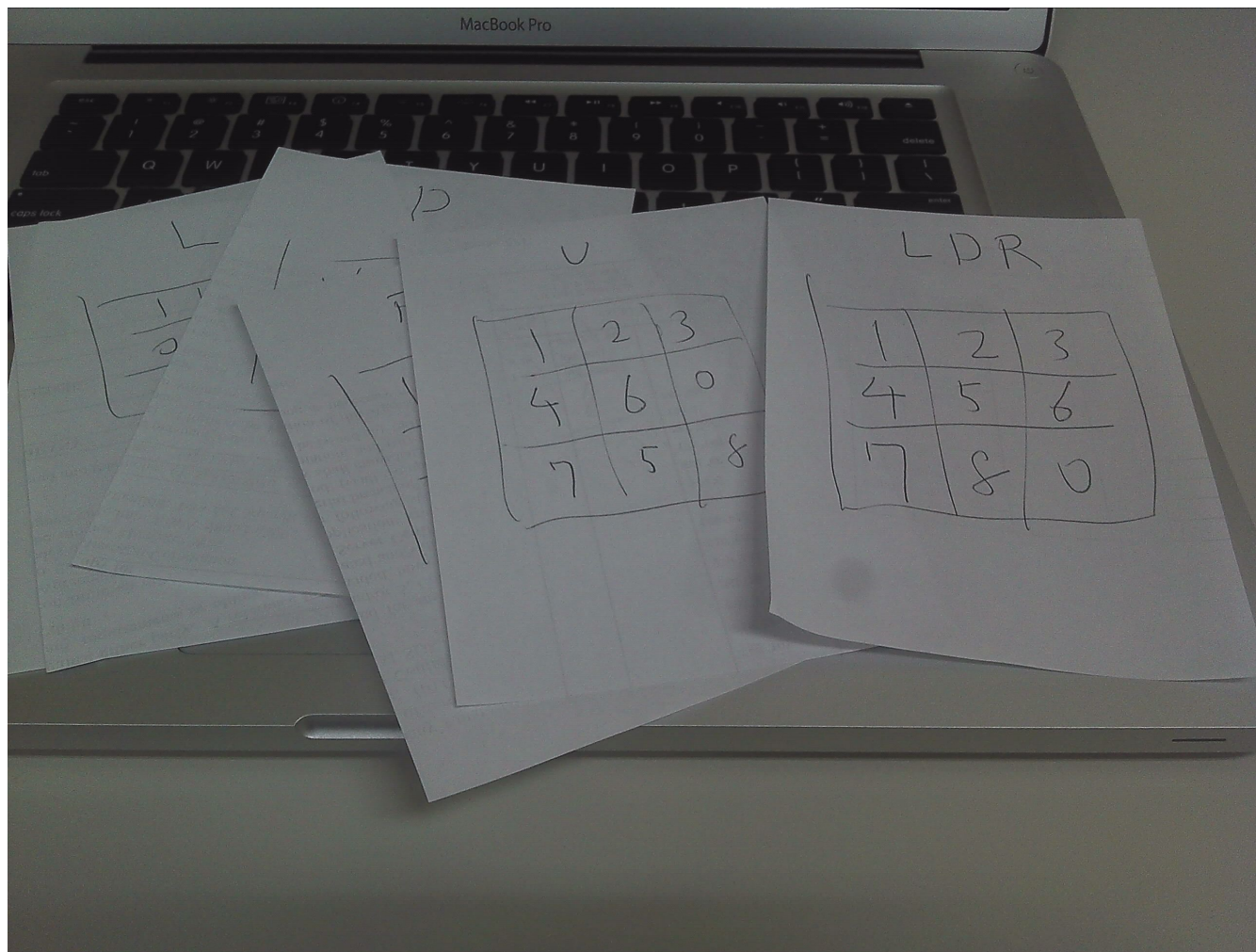
C-LIS

GDD2011 Tokyo - devQuiz答え合わせ大会 in 関西

2011/09/16

# スライドパズル発展の歴史

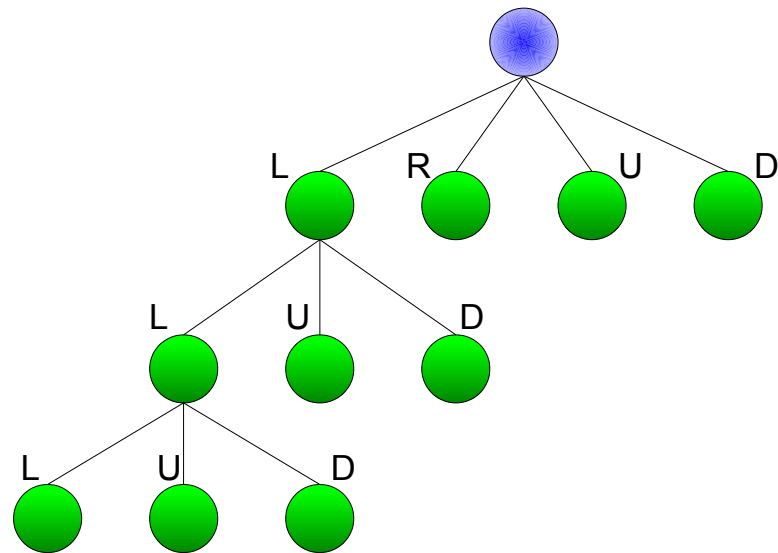
# 手動の時代



# プログラムの開発

- Javaで記述
- 解答は1問につき1つ、それぞれ別のファイルに書き出し  
集計は別のプログラムで

# 深さ優先探索



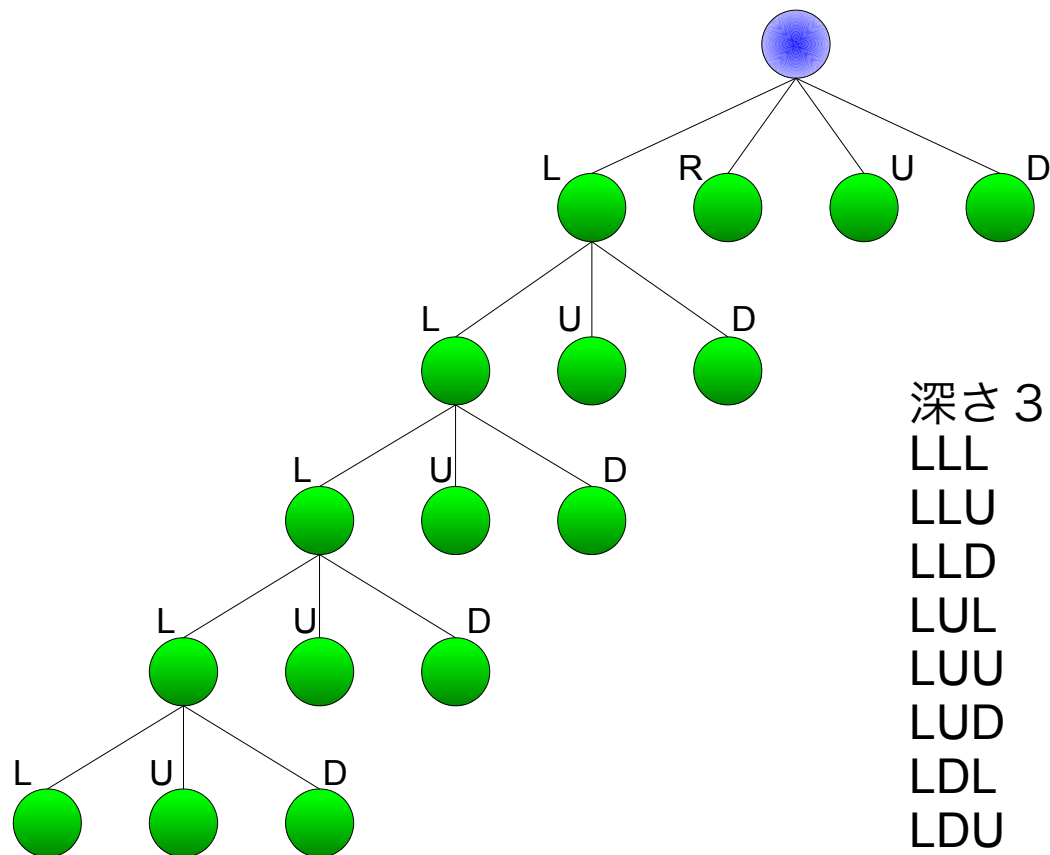
深さ 3

LLL  
LLU  
LLD  
LUL  
LUU  
LUD  
LDL  
LDU  
LDD  
...

何ステップで解答になるかが解らない

全探索の時代

# 反復深化 深さ優先探索



深さ 3

LLL  
LLU  
LLD  
LUL  
LUU  
LUD  
LDL  
LDU  
LDD  
...

深さ 5

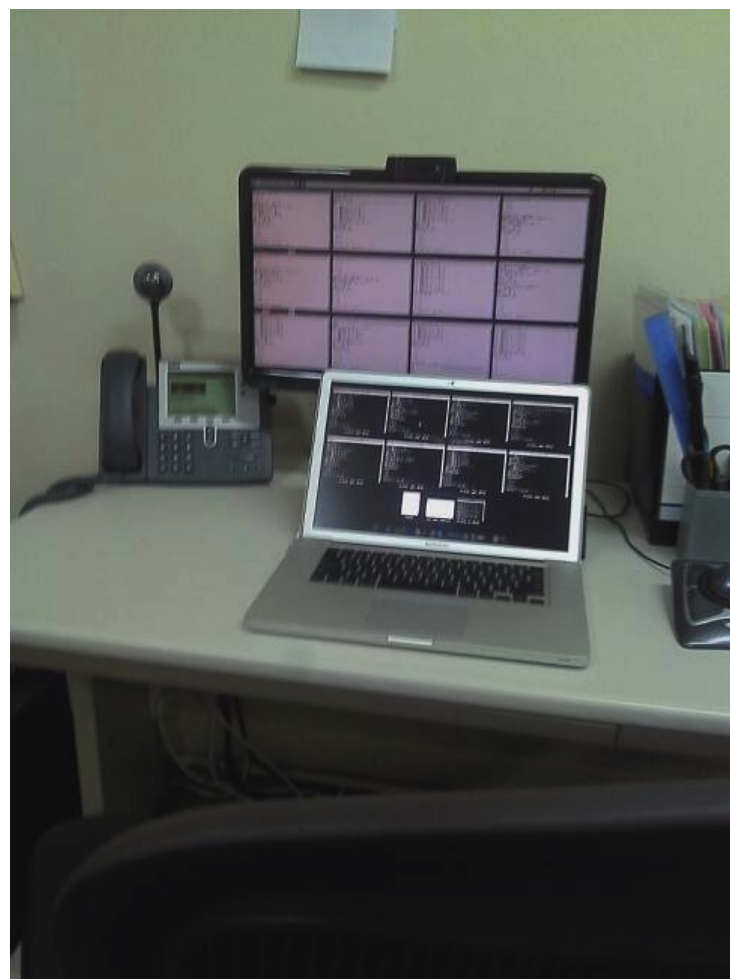
LLLLL  
LLLLU  
LLLLD  
LLLUL  
LLLUU  
LLLUD  
LLLDL  
LLLDU  
LLLDD  
...

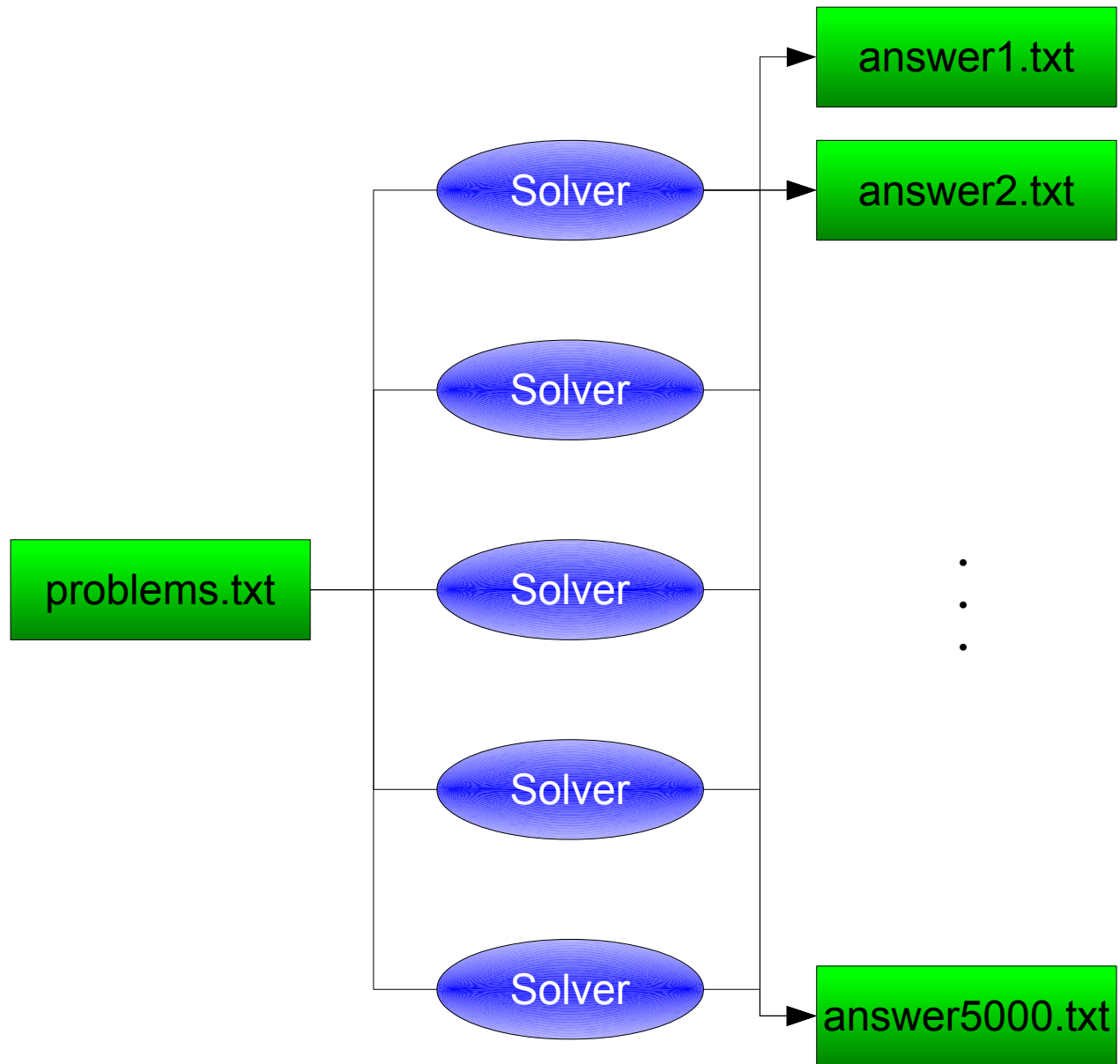
深さ 7

...



# 複数台のマシンを投入





## 解答ファイルの例

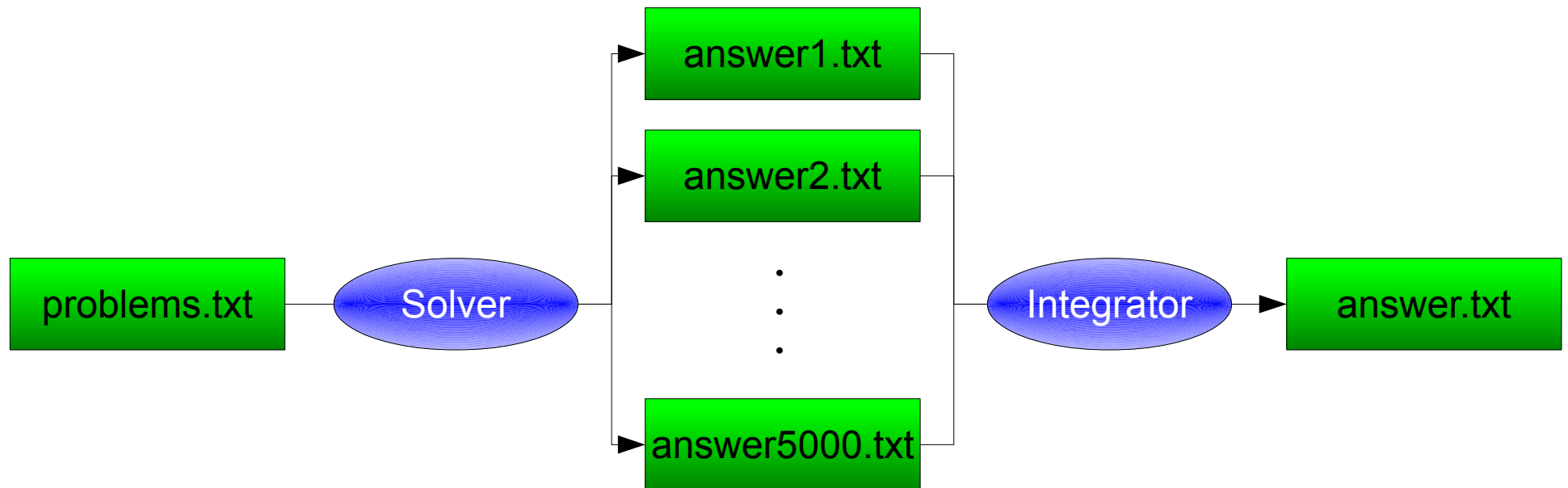
[illegible]

RDLDRULLDLURULDRDRRULURULLDRULLDRULLDRRRUULLDRR  
RDLDRULLDLURUULLDRDRRULULLDRDRURULLDRULLDRDLDRRURULLDRR  
RDLDRULLDLURUULLDRDRUULLDRDRURULLDRULLDRDLDRRULLDRULDR  
RDLDRULLDLURULLDRDRUULLDRDRURULLDRULLDRDLDRRRUULLDRULDR  
RDDLURULLLURDDRRULLLDRRUULLDDLUURDRURULLDDDRULURDUULLDLDR  
RDDLURULLDRRULLURDLDRRUULLDDLUURDRURULLDDDRULURDUULLDLDR  
DLDLURULRDRLDRUULLDRDLURDLURULLDRDRURULLDLDRRR  
DLDRURDLLLURRDLUULLURDLDRDUULLDRURULLDLDDRURUULLDDRR  
DLUULDRDRDLULLURRULULLDDRDRUULLDRUURULLDRUULLDDRURULDDR  
DRDLLLURRULLURDLDRDRURULULLDDRURULLDDRURUULLDLDRURDLDRR  
DRDLLLURRULLURDLDRDRURULULLDDRURULLDDRURUULLDLDRURDLDRR  
DRDLLLURULDRRULLDRDDLURDDLULURDRRUULLDRRULLURDDLULDRR  
DRDLLURDLLURRDLUULLURDDRULLDRULLDRRRUULLDRUULLDDR  
DRDLLURDLURRULLURDDRULLDRDDLURDDLULURDRRUULLDRUULLDDR  
DRDLLURDLURRULLURDDRULLDRDDLURDDLULURDRRUULLDRUULLDDR  
DRDLLURDLURULURDLDRRDLUULLDDRURULLDDLDRRUULLDRRUULLDDR  
DRDLLURDLURULURDLDRRDLUULLDDRURULLDDLDRRUULLDRRUULLDDR  
DRDLLURDLURULURDLDRRDLUULLDDRURULLDDLDRRUULLDRRUULLDDR  
DRDLLURDLURULURDLDRRDLUULLDDRURULLDDLDRRUULLDRRUULLDDR  
DRDLLURDLURULURDLDRRDLUULLDDRURULLDDLDRRUULLDRRUULLDDR  
DRDLULDRULURDDLURRDLUULLDRRULDRULLDRRRUULLDRUULLDDR  
DRDLURDLURDLUULLURDDLURRULLDRULLDRULLDRRRUULLDRUULLDDR  
DRDLULDRUULLURDDLURRDLUULLDRRULDRULLDRRRUULLDRUULLDDR  
DRDLURDLURDLURDDLURRULLDRRDLUULLDRULLDRRRUULLDRUULLDDR  
DRDLURULLURDLDRUULLDDRURULLDDLDRULLURDDRULURUULLDDR  
DRULLURDLDRDRUULLDDRULLDRULURDLURULLDRDLURRULLDDR  
DDRULLDLURULURDLDRDRULUULLDDRDLURURULLDLDRDRURULLDLDRRR  
DDRULLDRRULLURULLURDLDRUULLDDRURULLDLDDRURUULLDDRR  
DRULLULURDLDDRURULLLDRRUULLDDRURULLDDLURDDRURUULLDDR  
DDRULLULURDLDDRURULUULLDLDRUULLDRUURULLDLURDRUULLDDR  
DDRULLULURDLDDRULLDRRULUULLDRRUULLDDLDRRUULLDDLURDDR  
URDDLDRULLDRRULLDRRUULLDDRURULLDDRURULLDRRULLDRRULLDDR  
URDDLDRULLDLURDRUULLDDRULUULDRDRURULLDRULLLDRRULLDDR  
URDDLDRULLDLURDLDRRULLURDLDRULURULLDRULLDRRULLDRRULLDDR  
URDDLDRULLDRUULLDRRULLDRRULLDRULURULLDRULLDRRULLDRRULLDDR  
URDDLDRULLDRRULLDRRULLDRRULLDRRULLDRRULLDRRULLDRRULLDDR  
URDDLDRULLDRRULLDRRULLDRRULLDRRULLDRRULLDRRULLDRRULLDDR

# 集計用プログラム - Integrator

- 解答が正しいかチェック
- 手数制限を元に、各動き（LRUD）に重み付けを行う
- 重み付けを元に、解答ステップ数を計算
- 提出用ファイルを出力

# 構成



深さ 5 0 を越えると、全探索に限界

探索を打ち切る  
枝刈りの基準を探す

# Manhattan Distance

<u>4</u>	1	E	C
8	6	0	7
9	B	A	5
3	2	F	D

1	2	3	<u>4</u>
5	6	7	8
9	A	B	C
D	E	F	0



<u>3</u>	1	4	2
3	0	3	1
0	1	1	4
5	3	0	3

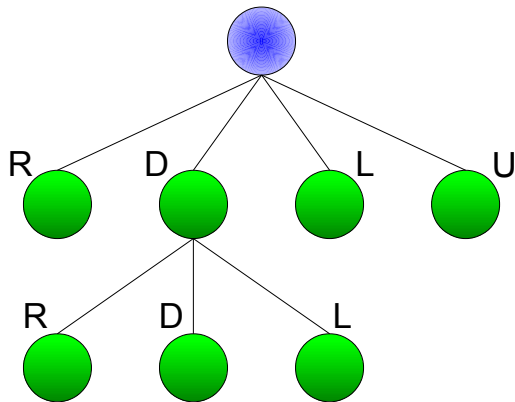
## Manhattan Distance

$$3 + 1 + 4 + 2 + 3 + 0 + 3 + 1 + 0 + 1 + 1 + 4 + 5 + 3 + 0 + 3 \\ = \underline{34}$$



大幅な速度アップを果たした  
2つの変更

# ランダムを導入



```
// int型の変数にactionと順番を保存する
// 0xFFFFFFFF
// 0x0000000F 0
// 0x000000F0 1
// 0x00000F00 2
// 0x0000F000 3
// 0x0F000000 length
```

```
public static int makeActionsList(char direction, Puzzle puzzle) {
    int actions = 0;
    int idx = 0;

    if (direction != Puzzle.DIRECTION_RIGHT && puzzle.canMoveLeft()) {
        actions = setAction(actions, Puzzle.DIRECTION_LEFT, idx);
        idx++;
    }

    if (direction != Puzzle.DIRECTION_LEFT && puzzle.canMoveRight()) {
        actions = setAction(actions, Puzzle.DIRECTION_RIGHT, idx);
        idx++;
    }

    if (direction != Puzzle.DIRECTION_UP && puzzle.canMoveDown()) {
        actions = setAction(actions, Puzzle.DIRECTION_DOWN, idx);
        idx++;
    }

    if (direction != Puzzle.DIRECTION_DOWN && puzzle.canMoveUp()) {
        actions = setAction(actions, Puzzle.DIRECTION_UP, idx);
        idx++;
    }

    actions = setActionLength(actions, idx);

    return shaffleActionsList(actions);
}
```

# 探索に時間制限

```
// 制限時間チェック  
if (mTimeLimit != -1 && (System.currentTimeMillis() - mTimeStart) > getTimeLimit()) {  
    return false;  
}
```

これで調子よく 3 3 0 0 問まで解き進む

そして、運命の9月3日

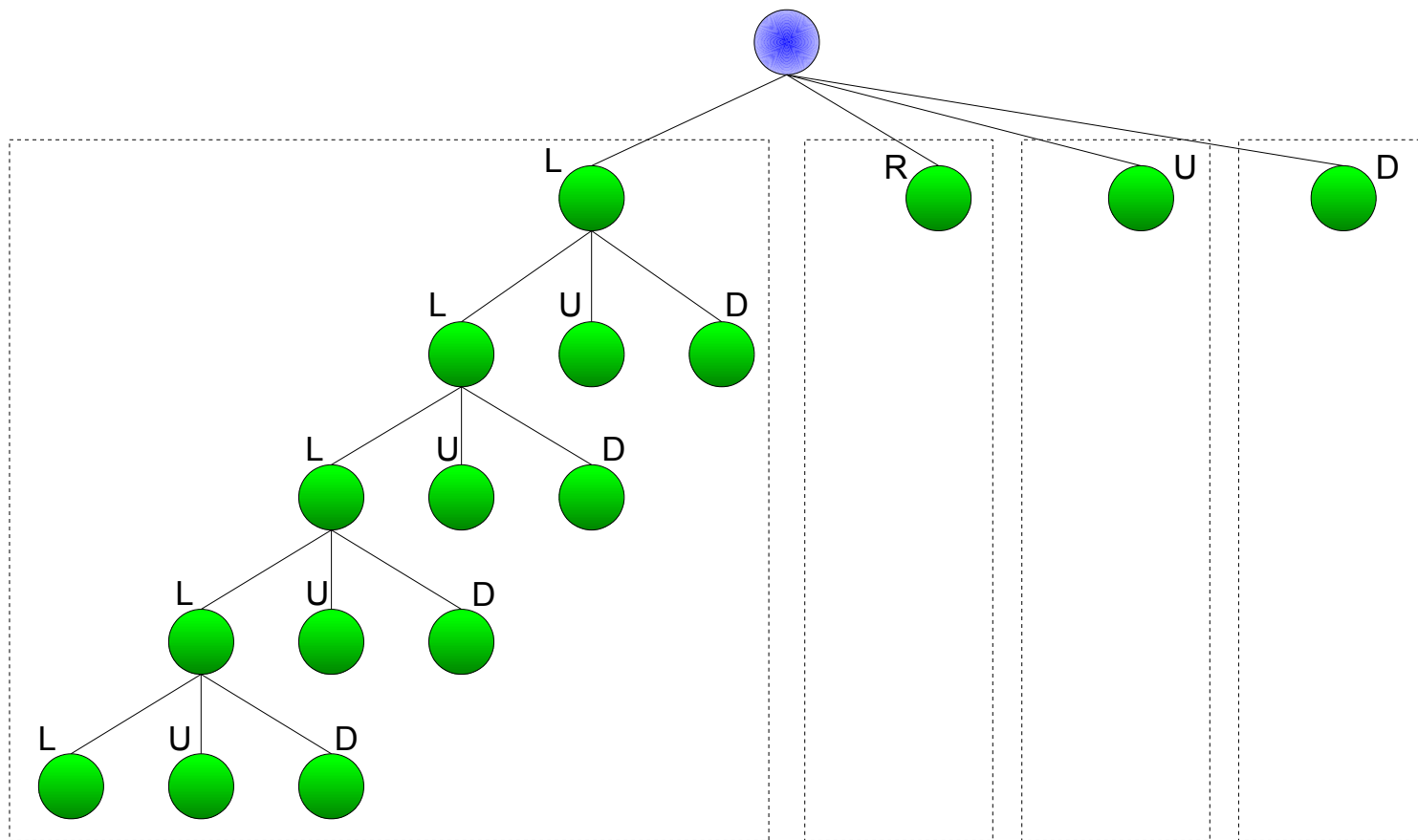
4900問以上解いてますが、なにか？



幅野郎との遭遇

さらなる回答数増加を目指して、  
細かい処理を最適化

# マルチスレッド化





# 探索範囲の制限

1	2	3	4
8	6	0	7
9	B	A	5
E	C	F	D

1	4	E	C
5	6	0	7
9	B	A	8
D	2	F	3

そして、「枝伸ばし」の導入

# 枝伸ばし

1	2	3	4
8	6	0	7
9	B	A	5
E	C	F	D

1	4	E	C
5	6	0	7
9	B	A	8
D	2	F	3

行または列が揃うと、それに応じて  
制限時間と探索の深さにボーナスを追加

# 枝伸ばし

```
private long getTimeLimit() {
    long time = mTimeLimit;

    // ボーナスを加算
    time += mTimeLimit
        * (Math.pow(mPuzzle.getFixedLines(), STEP_LIMIT_WEIGHT) + Math.pow(
            mPuzzle.getFixedCols(), STEP_LIMIT_WEIGHT));

    return time;
}

private long getStepLimit() {
    int step = mLimit;

    // ボーナスを加算
    step += mPuzzle.hi * (Math.pow(mPuzzle.getFixedLines(), STEP_LIMIT_WEIGHT) +
        mPuzzle.w * Math.pow(mPuzzle.getFixedCols(), STEP_LIMIT_WEIGHT));

    return step;
}
```

面白いように問題が解け、  
解答数は4 0 0 0問を突破

しかし...

この手法は解答までの手数が  
多くなる傾向

LDRULDRUULLUUURDRLDDRURDRUUULDLLDDRULLDRRLDRRULLURDLULDRURUURULDRDLDLLDRULU  
URULLDRURRDDLLULDDDDRURDRRUUULLLLURDRDDLURDDLURRUULURRDLURDLLULDRURDLLURRDDLLD  
DLDRULUURURDRDDDLLLURUULDRULURDRULDRRDDDLLUUUURDRLURDLDDDRRUUULLURDRULLDRRDD  
D  
LDRULDRUULLUUURDRLDDRURDRUUULDLLDDRULLDRRLDRRULLURDLULDRURUURULDRDLDLLDRULU  
URULLDRURRDDLLULDDDDRURDRRUUULLLLURDRDDLURDDLURRUULURRDLURDLLULDRURDLLURRDDLLD  
DLDRULUURURDRDDDLLLURUULDRULURDRULDRRDDDLLUUUURDRLURDLDDDRRUUULLURDRULRDD  
D  
LDRULDRUULLUUURDRLDDRURDRUUULDLLDDRULLDRRLDRRULLURDLULDRURUURULDRDLDLLDRULU  
URULLDRURRDDLLULDDDDRURDRRUUULLLLURDRDDLURDDLURRUULURRDLURDLLULDRURDLLURRDDLLD  
DLDRULUURURDRDDDLLLURUULDRULURDRULDRRDDDLLUUUURDRLURDLDDDRRUUULLURDRULRDD  
D  
LDRULDRUULLUUURDRLDDRURDRUUULDLLDDRULLDRRLDRRULLURDLULDRURUURULDRDLDLLDRULU  
URULLDRURRDDLLULDDDDRURDRRUUULLLLURDRDDLURDDLURRUULURRDLURDLLULDRURDLLURRDDLLD  
DLDRULUURURDRDDDLLLURUULDRULURDRULDRRDDDLLUUUURDRLURDLDDDRRUUULLURDRULRDD  
D  
LDRULDRUULLUUURDRLDDRURDRUUULDLLDDRULLDRRLDRRULLURDLULDRURUURULDRDLDLLDRULU  
URULLDRURRDDLLULDDDDRURDRRUUULLLLURDRDDLURDDLURRUULURRDLURDLLULDRURDLLURRDDLLD  
DLDRULUURURDRDDDLLLURUULDRULURDRULDRRDDDLLUUUURDRLURDLDDDRRUUULLURDRULRDD  
D

手数制限に引っかかりそうに。



# RouteShorter の登場

- 既に解答を得られた問題を再び探索
- 現在得られている最短ステップより下の制限で探索を行う

```
boolean targetFlg = false;
do {
    // 回答一覧
    Answer[] list = Answer.getAnswerList(sProblems);

    for (int i = 0; i < list.length; i++) {
        Answer answer = list[i];

        if (isTarget(answer)) {
            targetFlg = true;

            String map = problems.get(answer.getNumber() - 1);
            Puzzle puzzle = new Puzzle(answer.getNumber(), map);

            int maxStep = answer.getMinStep();
            if (maxStep >= 0) {
                int initStep = maxStep - sCutStep;
                if ((initStep % 2) == 0) {
                    initStep += ((answer.getMinStep() % 2 == 1) ? 1 : 0);
                }

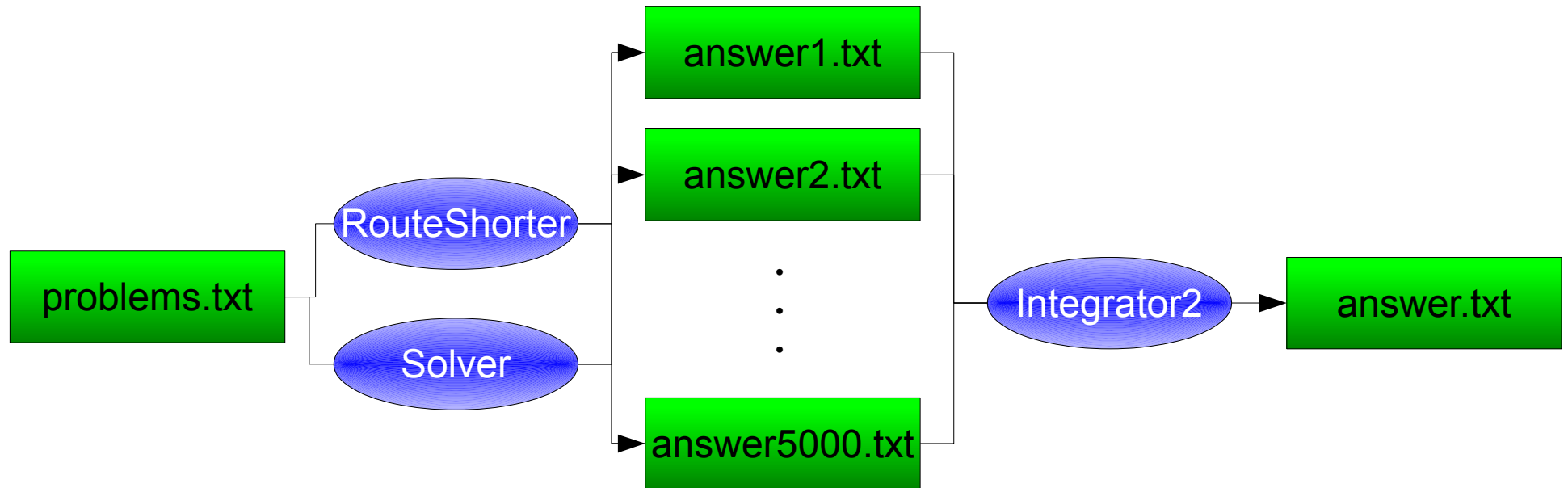
                // パズルの実行
                boolean result = false;
                do {
                    result = doPuzzle(puzzle, initStep, maxStep);
                    success += result ? 1 : 0;
                    initStep += 2;
                } while (!result && initStep < maxStep);
            }
        }
    }
} while (targetFlg);
```

LDRULDRUULLUUURRDLLDDRURDRUUULDLLDDRULLDRRDLDRRULLURRDLULDRURUURULDRDLDLLDRRULU  
URULLDRURRDDLLULDDDDRURDRRUUULLLLURDRDDLLURDDLURRUULURRRDLURDLLULDRURDLLURRRDLLD  
DLDRULUURURDRDDDLLLURUULDRULURDRULDRRDDDLLUUUURRDLLURRDLLDDDRRUUULLURDRULLDRRDD  
D  
LDRULDRUULLUUURRDLLDDRURDRUUULDLLDDRULLDRRDLDRRULLURRDLULDRURUURULDRDLDLLDRRULU  
URULLDRURRDDLLULDDDDRURDRRUUULLLLURDRDDLLURDDLURRUULURRRDLURDLLULDRURDLLURRRDLLD  
DLDRULUURURDRDDDLLLURUULDRULURDRULDRRDDDLLUUUURRDLLURRDULDDDDRURUULLURRDURDDD  
D  
LDRULDRUULLUUURRDLLDDRURDRUUULDLLDDRULLDRRDLDRRULLURRDLULDRURUURULDRDLDLLDRRULU  
URULLDRURRDDLLULDDDDRURDRRUUULLLLURDRDDLLURDDLURRUULURRRDLURDLLULDRURDLLURRRDLLD  
DLDRULUURURDRDDDLLLURUULDRULURDRULDRRDDDLLUUUURDLURRDLLDDDRRUUUULDURRDLLURDRUL  
DRULDRDDD  
LDRULDRUULLUUURRDLLDDRURDRUUULDLLDDRULLDRRDLDRRULLURRDLULDRURUURULDRDLDLLDRRULU  
URULLDRURRDDLLULDDDDRURDRRUUULLLLURDRDDLLURDDLURRUULURRRDLURDLLULDRURDLLURRRDLLD  
DLDRULUURURDRDDDLLLURUULDRULURDRULDRRDDDLLUUUURDLURRDLLDDDRRUUUULDURRDLLURRDD  
D  
LDRULDRUULLUUURRDLLDDRURDRUUULDLLDDRULLDRRDLDRRULLURRDLULDRURUURULDRDLDLLDRRULU  
URULLDRURRDDLLULDDDDRURDRRUUULLLLURDRDDLLURDDLURRUULURRRDLURDLLULDRURDLLURRRDLLD  
DLDRULUURURDRDDDLLLURUULDRULURDRULDRRDDDLLUUUURDLURRDULDDDDRURUUUULDURDRULDRDD  
D

ULLDRRRDRRUUUULDLDLULLDDRDRUULDLDUULLUUURRDDRUURDLULLDLRRULLDRRDRDDLURDLLULUUR  
RUULDDRURRRDDDDLLLUURULURDRRURDDDDLLUUUURRDLLDDDRRUUULURDDDD  
ULLDRRRDRRUUUULDLDLULLDDRDRUULDLDUULLUUURRDDRUURDLULLDLRRULLDRRDRDDLURDLLULUUR  
RUULDRRRRRDDDDLLLUUURDLURURURDDDDLLUUUURRDLLDDDRRUUULURDDDD  
ULLDRRRDRRUUUULDLDLULLDDRDRUULDLDUULLUUURRDDRUURDLULLDLRRULLDRRDRDLDRULDLULUUR  
RUULDDRURRRDDDDLLLUURULURDRRURDDDDLLUUUURRDLLDDDRRUUULURDDDD  
ULLDRRRDRRUUUULDLDLULLDDRDRUULDLDUULLUUURRDDRUURDLULLDLRRULLDRRDRDLDRULDLULUUR  
RUULDRRRRRDDDDLLLUUURDLURURURDDDDLLUUUURRDLLDDDRRUUULURDDDD

# Integrator2 の開発

- 各ステップの重み付けを改良
- ただし、最初に全ての解答をメモリに格納するため低速



順調に残りの手数が増加

4 8 0 0 問

なかなか解けない

残り200問

# キャッシュの導入

- 探索の過程（盤面）をメモリに保存して同じ盤面があれば以降の探索を回避
- キャッシュが大きくなると速度が低下
- ハッシュ衝突の可能性

```
synchronized (cache) {  
    if (cache.contains(mPuzzle)) {  
        cacheHit++;  
        return false;  
    }  
    if (cache.size() < cacheSize) {  
        cache.add(mPuzzle);  
    }  
}
```

# Compressor の開発

- 。 既存の解答をもとに盤面をチェック。  
同じ盤面で短いルートがあればそちらに置き換え（ショートカット）する

```
for (int i = 0; i < len; i++) {
    move(r.charAt(i));

    if (cache.containsKey(mPuzzle)) {
        String rt = cache.get(mPuzzle);
        if (rt.length() < route.length()) {
            cache.put(mPuzzle.duplicate(), route.toString());

            // ルートをより短いもので上書きする
            route.delete(0, route.length());
            route.append(rt);
        } else {
            // キャッシュから削除
            cache.remove(mPuzzle);
        }
    } else {
        cache.put(mPuzzle.duplicate(), route.toString());
    }
}
```



Compressorの効果は絶大  
使用手数が最大で50%圧縮

4 9 9 9 問の解答を得る

# パズルの終焉 - 立ちはだかった1052

D	8	2	3	6	B
K	E	=	=	5	4
1	0	P	V	G	J
7	W	=	F	=	C
X	=	Y	H	U	I
R	S	M	T	Z	O

ギリギリまで計算を続けるが、  
解答を得られず

devQuiz終了...

Result - Google Developer D x

gdd-2011-quiz-japan.appspot.com/result

Google Developer Day 2011

keiji.ariyama | ログアウト

Quiz トップ  
よくあるご質問  
採点結果

### Result

問題	小問	結果	得点	詳細	提出時刻
スライドバズル	1	採点済み	49.99	<a href="#">Detail</a>	2011-09-11 11:06:24.670533
スライドバズル	1	採点済み	49.97	<a href="#">Detail</a>	2011-09-11 02:53:18.871192
スライドバズル	1	採点済み	49.95	<a href="#">Detail</a>	2011-09-10 18:07:22.647628
スライドバズル	1	採点済み	49.94	<a href="#">Detail</a>	2011-09-10 16:54:19.516600
スライドバズル	1	採点済み	49.87	<a href="#">Detail</a>	2011-09-09 11:28:06.651725
スライドバズル	1	採点済み	49.56	<a href="#">Detail</a>	2011-09-08 08:17:58.740772
スライドバズル	1	採点済み	49.5	<a href="#">Detail</a>	2011-09-08 07:16:18.754302
スライドバズル	1	採点済み	48.97	<a href="#">Detail</a>	2011-09-07 22:50:34.445873
スライドバズル	1	採点済み	47.5	<a href="#">Detail</a>	2011-09-07 17:27:02.897868
スライドバズル	1	採点済み	45.13	<a href="#">Detail</a>	2011-09-07 09:26:43.428027
スライドバズル	1	採点済み	43.96	<a href="#">Detail</a>	2011-09-07 06:50:21.913247
スライドバズル	1	採点済み	5.45	<a href="#">Detail</a>	2011-09-06 21:41:39.523063
スライドバズル	1	採点済み	42.03	<a href="#">Detail</a>	2011-09-06 09:20:41.760219
スライドバズル	1	採点済み	40.03	<a href="#">Detail</a>	2011-09-05 16:07:34.084825
スライドバズル	1	採点済み	0.57	<a href="#">Detail</a>	2011-09-05 15:35:48.452773
スライドバズル	1	不正解	0.0	<a href="#">Detail</a>	2011-09-05 15:33:55.513061
スライドバズル	1	採点済み	36.72	<a href="#">Detail</a>	2011-09-04 02:44:29.837528
スライドバズル	1	採点済み	2.4	<a href="#">Detail</a>	2011-09-03 16:17:18.759445
スライドバズル	1	採点済み	2.4	<a href="#">Detail</a>	2011-09-03 16:15:54.072435
スライドバズル	1	採点済み	35.17	<a href="#">Detail</a>	2011-09-02 19:11:07.233027
スライドバズル	1	採点済み	33.51	<a href="#">Detail</a>	2011-09-02 08:36:47.000038
スライドバズル	1	採点済み	31.48	<a href="#">Detail</a>	2011-09-02 00:14:19.494263
スライドバズル	1	採点済み	26.8	<a href="#">Detail</a>	2011-09-01 02:33:50.328838
スライドバズル	1	採点済み	25.01	<a href="#">Detail</a>	2011-08-31 15:29:13.715120
スライドバズル	1	採点済み	24.0	<a href="#">Detail</a>	2011-08-31 11:28:43.059135
スライドバズル	1	採点済み	21.81	<a href="#">Detail</a>	2011-08-31 02:26:50.725209
スライドバズル	1	採点済み	20.17	<a href="#">Detail</a>	2011-08-30 20:10:04.784135
スライドバズル	1	不正解	0.0	<a href="#">Detail</a>	2011-08-30 20:07:04.062266
スライドバズル	1	採点済み	19.71	<a href="#">Detail</a>	2011-08-30 17:43:10.771860
スライドバズル	1	不正解	0.0	<a href="#">Detail</a>	2011-08-30 17:41:10.808067
スライドバズル	1	採点済み	18.35	<a href="#">Detail</a>	2011-08-30 11:29:36.435798
スライドバズル	1	採点済み	18.14	<a href="#">Detail</a>	2011-08-30 10:17:28.225749
スライドバズル	1	採点済み	17.65	<a href="#">Detail</a>	2011-08-30 02:13:27.958802

# さいごに

- 最初から完璧を目指すのではなく、調べながらの改良は快感
- 最後の1問が解けなかったのは、重い事実
- 次回挑戦時には、経路選択時の重み付けをきちんと行いたい

# 質疑応答

本資料は、**有限会社シーリス**の著作物であり、  
クリエイティブコモンズの表示-非営利-継承 3.0 Unported ライセンスの元で公開しています。





ご清聴ありがとうございました。