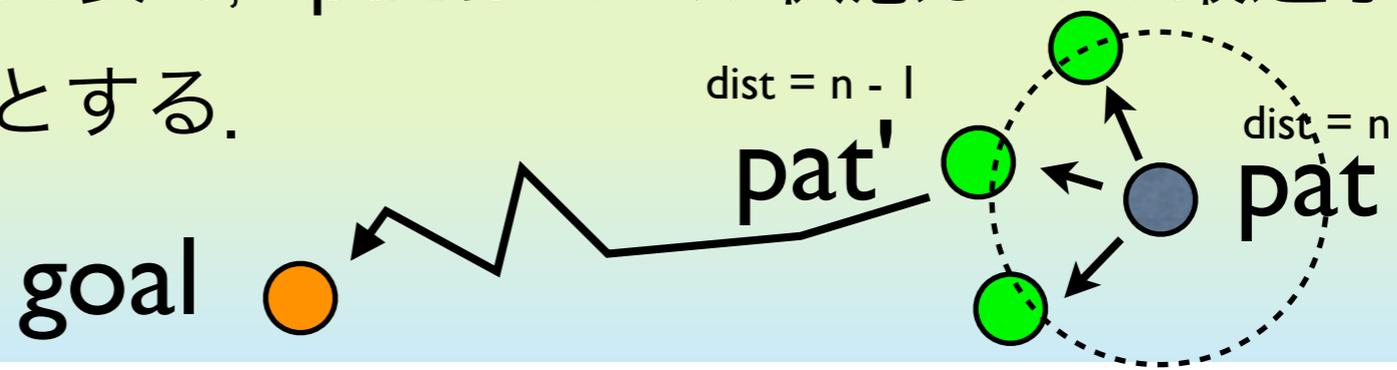


本日の課題

本日解説したハッシュ表を用いた8パズルの探索で、任意の状態 pat からゴール状態へ至る最適な手順をハッシュ表を用いて計算するプログラムを作れ。ハッシュ表を用いればあるパターン pat から1ステップで移動可能なパターン pat' でゴール状態までの距離が1少なくなるものが必ず存在する。そのような状態をゴール状態に達するまで順に調べていけば、最適な動かし方がわかる。 `kadai(hash_table, pat)` がコマの移動を表す文字列 "rulull..." を返すようにせよ。ただし、`hash_table` は講義で説明したハッシュ表で、 pat はゴール状態までの最適手順が知りたいパターンとする。



解答例 (1)

これまで説明した関数を組み合わせて用いれば作ることができる。
手順は問題で示した通りである。

- 1.与えられた状態sから最終状態Fまでの最適ステップ数dを調べる。
- 2.一つ動かしてみてステップ d - 1になる方向を探す（これは必ずあるはず）。
- 3.そちらの方向にsを移動させ、それを記録する。

```
function kadai(pat_table, pat){
    hash_table = pat_table
    var output = "";
    var ddd = [UP, DOWN, RIGHT, LEFT]
    var d = find_pat(pat)
    while (d > 0){
        for (var i = 0; i < 4; i++){
            pat_new = move(pat, ddd[i])
            var d1 = find_pat(pat_new)
            if (d1 == d - 1) {
                d = d1 ;
                pat = pat_new
                output += dir[i]
                break;
            }
        }
    }
    return output
}
```

解答例 (2)

3

実際に8パズルの実行環境（ハッシュ表の計算結果を付加したものの）の上で実行した結果を以下に示す。探索時間は0～1ms程度（おそらく多くの場合1ms未満）となっている。

http://osami.s280.xrea.com/Algo_Data2015/Interpreter/Interpreter5-hash.html

ハッシュ表を用いた最適解の探索

すでに pat_table にハッシュ表が入っているとする。すべて授業で説明した関数を用いて以下のように実現することができる。答えをいちいち探索するわけではないので、瞬時に答えが得られる。

結果消去 プログラム消去 実行

```
function kadai(pat_table, pat){
  hash_table = pat_table
  var output = "";
  var ddd = [UP, DOWN, RIGHT, LEFT]
  var d = find_pat(pat)
  while (d > 0){
    for (var i = 0; i < 4; i++){
      pat_new = move(pat, ddd[i])
      var d1 = find_pat(pat_new)
      if (d1 == d - 1) {
        d = d1 ;
        pat = pat_new
        output += dir[i]
        break;
      }
    }
  }
  return output
}
```

入力 :

出力 :
initial state : [0,8,6,1,7,3,4,2,5]
move : ululdrdluurdlurrdldrulurdd (26 moves)
Time for finding an optimal solution: 0 ms

	1	2
3	4	5
6	7	8