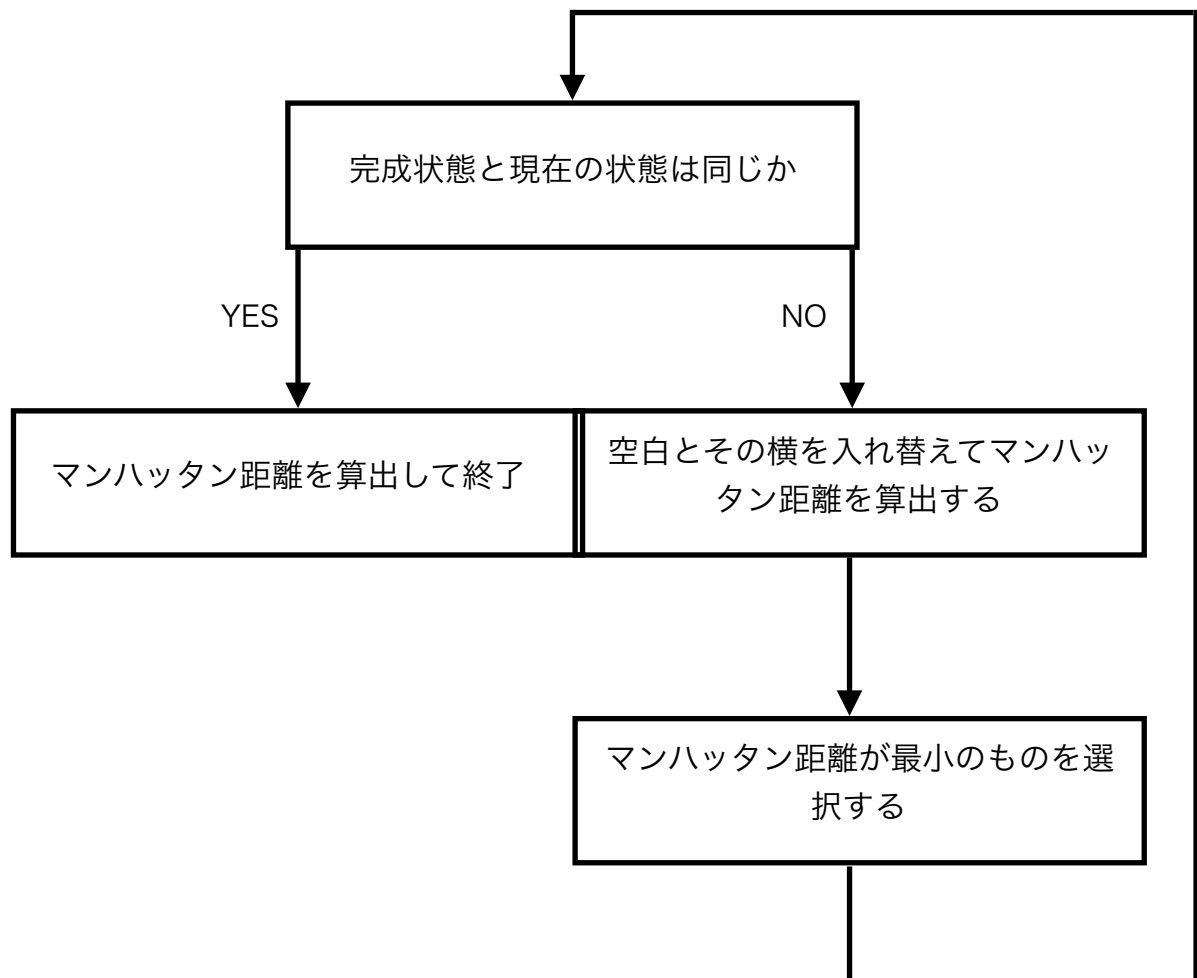


1. フローチャート



2. ソースコード

```
# -*- coding: utf-8 -*-
import math
import numpy as np
import sys
manhattankyori = 1
shoki = 1
a = 0
Alist = [[1, 2, 3], [4, 5, 6], [7, 8, 0]]
Klist1 = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
Klist2 = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
Klist3 = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
Klist4 = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]

#マンハッタン距離計算
def kyori(k, l):
    t = 0
    global manhattankyori
    global shoki
    global Qlist
    Qlist = [[8, 1, 5], [2, 0, 4], [6, 3, 7]]
    # 空白0の入れ替え
    for i in range(0, 3):
        for j in range(0, 3):
            if Qlist[i][j] == 0 and t == 0:
                e = Qlist[i + k][j + l]
                r = Qlist[i][j]
                Qlist[i + k][j + l] = r
                Qlist[i][j] = e
                for o in range(0, 3):
                    for p in range(0, 3):
                        sys.stdout.write(str(Qlist[o][p]))
                print()
                print()
                t = 1
    mm = 0
    for i in range(0, 3):
        for j in range(0, 3):
            for x in range(0, 3):
                for y in range(0,3):
```

```
# 答えとなるリストと比較してマンハッタン距離を算出
if Qlist[i][j] != 0 and Qlist[i][j] == Alist[x][y]:
    xm = x - i
    ym = y - j
    xm = abs(xm)
    ym = abs(ym)
    m = xm + ym
    m = abs(m)
    mm = mm + m
    manhattankyori = mm
```

```
k = 0
```

```
l = 0
```

```
kyori(k, l)
```

```
shoki = manhattankyori
```

```
# 入れ替えたリストを保存
```

```
# 縦方向の空白の入れ替え
```

```
l = 0
```

```
for k in range(-2, 0):
```

```
    kyori(k,l)
```

```
    if k == -2:
```

```
        sita = manhattankyori
```

```
        Klist1 = Qlist
```

```
    if k == -1:
```

```
        ue = manhattankyori
```

```
        Klist2 = Qlist
```

```
# 横方向の空白の移動
```

```
k = 0
```

```
for l in range(-2, 0):
```

```
    kyori(k,l)
```

```
    if l == -2:
```

```
        migi = manhattankyori
```

```
        Klist3 = Qlist
```

```
    if l == -1:
```

```
        hidari = manhattankyori
```

```
        Klist4 = Qlist
```

```
a = a + 1
```

```
Mlist = [int(sita), int(ue), int(migi), int(hidari)]
```

```
Mlist.sort()
```

```
print ("初期マンハッタン距離" + str(int(shoki)))
```

```
f = open('output.txt','a')
```

```
f.write("初期マンハッタン距離" + str(int(shoki)) + "\n")
```

```
shoki = Mlist[1]
print("空白を上に移動したマンハッタン距離" + str(int(ue)))
f.write("空白を上に移動したマンハッタン距離" + str(int(ue)) + "\n")
print(Klist2)
f.write(str(list(Klist2)) + "\n")
print("空白を下に移動したマンハッタン距離" + str(int(sita)))
f.write("空白を下に移動したマンハッタン距離" + str(int(sita)) + "\n")
print(Klist1)
f.write(str(list(Klist1)) + "\n")
print("空白を左に移動したマンハッタン距離" + str(int(hidari)))
f.write("空白を左に移動したマンハッタン距離" + str(int(hidari)) + "\n")
print(Klist4)
f.write(str(list(Klist4)) + "\n")
print("空白を右に移動したマンハッタン距離" + str(int(migi)))
f.write("空白を右に移動したマンハッタン距離" + str(int(migi)) + "\n")
print(Klist3)
f.write(str(list(Klist3)) + "\n")
print("マンハッタン距離" + str(list(Mlist)))
f.write("マンハッタン距離" + str(list(Mlist)) + "\n")
print("最小のマンハッタン距離" + str(int(shoki)))
f.write("最小のマンハッタン距離" + str(int(shoki)) + "\n")
print("移動回数" + str(int(a)) + "回")
f.write("移動回数" + str(int(a)) + "回\n")
f.close()
```

3. 実行環境

Mac Book Air Mid 2013

Mac OS Sierra ver. 10.12.6

Python 3.6.1 (default, Apr 4 2017, 09:40:21)

[GCC 4.2.1 Compatible Apple LLVM 8.1.0 (clang-802.0.38)] on darwin

4. 実行結果

初期マンハッタン距離18

空白を上に移動したマンハッタン距離19

[[8, 0, 5], [2, 1, 4], [6, 3, 7]]

空白を下に移動したマンハッタン距離17

[[8, 1, 5], [2, 3, 4], [6, 0, 7]]

空白を左に移動したマンハッタン距離17

[[8, 1, 5], [0, 2, 4], [6, 3, 7]]

空白を右に移動したマンハッタン距離17

[[8, 1, 5], [2, 4, 0], [6, 3, 7]]

マンハッタン距離[17, 17, 17, 19]

最小のマンハッタン距離17

移動回数1回