人工知能　課題

5CS 42　納谷　知郎

1，実装環境

Visual studio 2017 C++

Windows 10 HOME 64bit

2，アルゴリズムの説明

アルゴリズムのフローチャートを以下の図１に示す．まず空きマスに隣り合う数字のマンハッタン距離の総和を求める．その中で最も小さいマンハッタン距離総和を持つ値を空きマスに移動させる．これを繰り返しパズルがゴール状態と同様になったら処理を終了する．

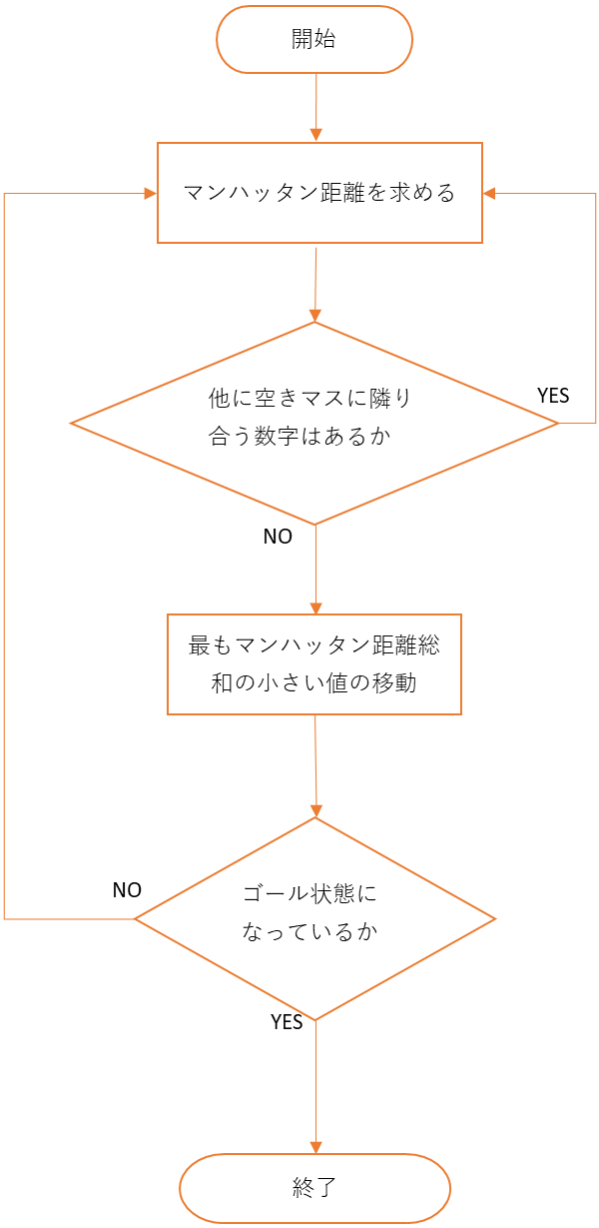

図 1　アルゴリズムのフローチャート

3，ソースコード

作成したプログラムを以下の図 2 に示す

マンハッタン距離は求められるがゴール状態までたどり着けなかった.

```c
#include <stdio.h>
#include <stdlib.h>
main() {
        int   s[3][3] = {8, 1, 5,
                         2, 0, 4,
                         6, 3, 7}     //初期(現在)状態
            ,g[3][3] = {1, 2, 3,
                         4, 5, 6,     //ゴール状態
                         7, 8, 0}     //空きマス：0
             ,w = 0                   //while
             ,c = 2                   //マンハッタン距離の例外処理用
             ,i,j                     //タイルの座標
             ,a                       //空きマス(0)の上下左右
             ,ai = 0,aj = 0     //一時保存する座標(マンハッタン距離の小さい座標)
             ,m = 20,ms = 0,mg = 0    //マンハッタン距離
             ,si,sj                    //距離を求めるとき    現在
             ,gi,gj;                   //に使用する座標      ゴール
        FILE *file;
        fopen_s(&file,"Manhattan.txt","w");
        do{
         for(i = 0; i<3; i++){
          for(j = 0; j<3; j++){ //空きマスを探索
           if(s[i][j]==0){
            printf("座標 (%d,%d) に空きマスを検出¥n",i+1,j+1,s[i][j]);
            for(a = -1; a<2; a = a+2){      //空きマスの上下左右に値が存在するか
             printf("空きマスから見て %d 方向を調べる¥n",a);
             if(s[i+a][j]>0&&s[i+a][j]<10&&i+1+a<4&&i+1+a > 0){ //上下の値
              printf("(%d,%d)= %d¥n",i+1+a,j+1,s[i+a][j]);
              c--;
              //値があるとき移動する
              s[i][j] = s[i+a][j];
              s[i+a][j] = 0;
              //マンハッタン距離を出す
              for(si = 0; si<3; si++){
               for(sj = 0; sj<3; sj++){ //現在の状態
                for(gi = 0; gi<3; gi++){
                 for(gj = 0; gj<3; gj++){ //ゴール
```

```
                if(s[si][sj]==g[gi][gj]&&s[si][sj]!=0){
                 //|現在の座標|-|ゴール時の座標|（絶対値）
                 mg = mg+abs(si-gi)+abs(sj-gj);
                }
               }
              }
             }
            }
    printf("この座標のマンハッタン距離は距離は%dです。¥n¥n",mg);
    s[i][j+a] = s[i][j];
    s[i][j] = 0;
         }
    if(ms == 0){
             ms = 21;
    }
    if(mg == 0){
             mg = 21;
    }
    if(c<2){
    //最も小さいマンハッタン距離を選ぶ
     if(mg<=ms && mg<=m){
      aj = a;
      ai = 0;
      m = mg;
    } else if(ms<mg && ms<=m){
      ai = a;
      aj = 0;
      m = ms;
    }
    }
    c = 2;
    printf("ms = %d¥nmg = %d¥n m = %d¥n",ms,mg,m);
    printf("ai = %d¥naj = %d¥n",ai,aj);
    ms = 0;
    mg = 0;
  }
```

```
//0(空きマス)に最もマンハッタン距離が小さくなった値を入れる
if(ai==0){
 s[i][j] = s[i][j+aj];
 s[i][j+aj] = 0;
 printf("s[%d][%d] = %d¥n",i+1,j+1,s[i][j]);
 } else if(aj==0){
 s[i][j] = s[i+ai][j];
 s[i+ai][j] = 0;
 printf("s[%d][%d] = %d¥n",i+1,j+1,s[i][j]);
}
fprintf(file,"現在の座標は¥n"
         "[%d][%d][%d]¥n"
         "[%d][%d][%d]¥n"
         "[%d][%d][%d]¥n¥n"
         ,s[0][0],s[0][1],s[0][2]
         ,s[1][0],s[1][1],s[1][2]
         ,s[2][0],s[2][1],s[2][2]);
ms = 0;
mg = 0;
m = 20;
                                        }
                                }
                        }
         } while(w==0);
         fclose(file);
}
```

図 2　作成したプログラム

4，実行結果

```
現在の座標は
[8][1][5]
[2][4][0]
[6][3][7]

現在の座標は
[8][1][0]
[2][4][5]
[6][3][7]

現在の座標は
[8][1][5]
[2][4][0]
[6][3][7]

現在の座標は
[8][1][0]
[2][4][5]
[6][3][7]
//以下無限ループ
```