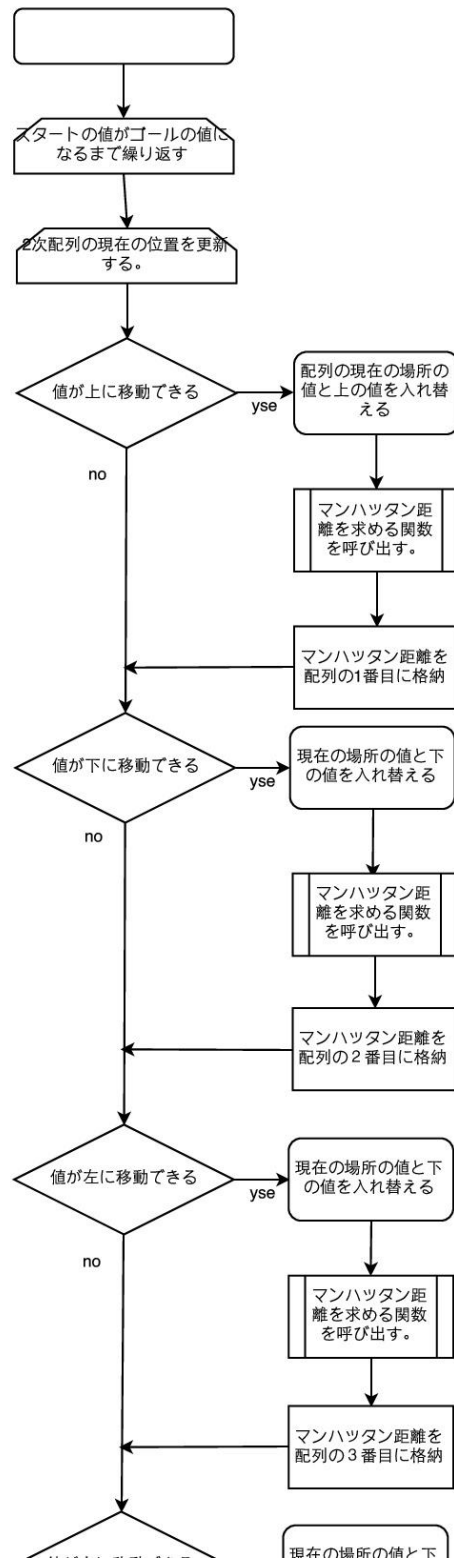
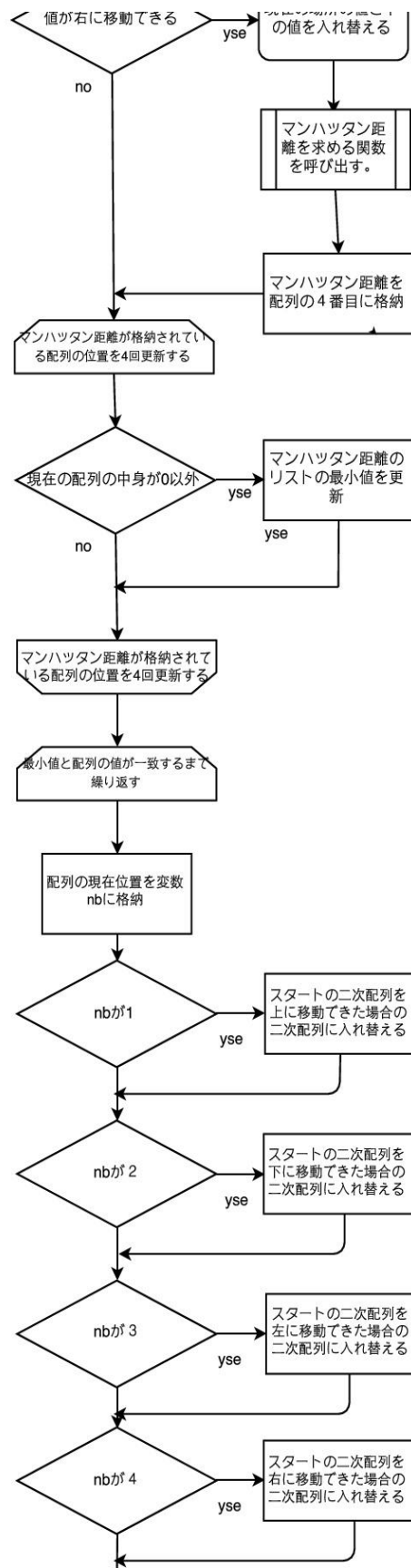


1 フローチャート





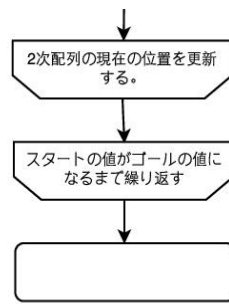


図 1 main 文フローチャート



図 2 マンハッタン距離を求める関数のフローチャート

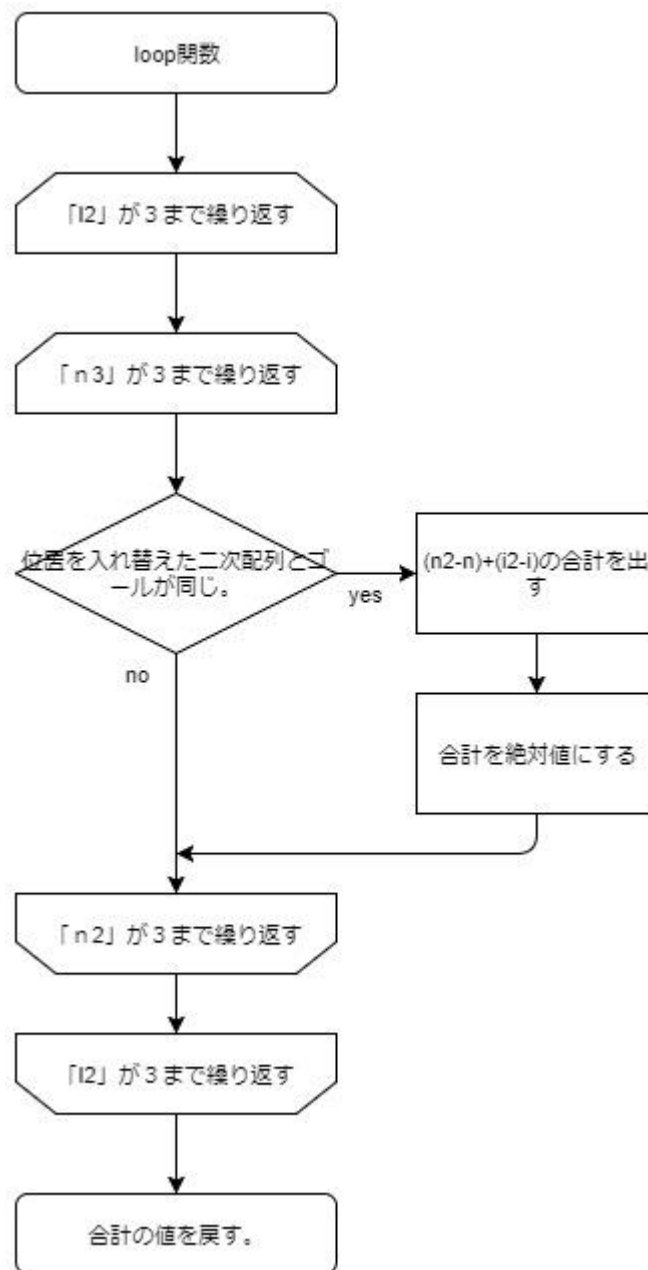


図 3 loop 関数のフローチャート

## 2 プログラム

```

#include<stdio.h>
#include<string.h>
#include <stdlib.h>
#include <math.h>
#define N
int s[3][3] = { //スタート
               { 8, 1, 5 },

```

```

        { 2, 0, 4},
        { 6, 3, 7}}};

int g[3][3] = { //ゴール
        { 1, 2, 3 },
        { 4, 5, 6 },
        { 7, 8, 0 }}};

int stac[3][3];
int stac1[3][3];
int stac2[3][3];
int stac3[3][3];
int stac4[3][3];

int i,n,i2,n2,mancount,minb,nb=0;
int mini=100;//最小値
int manstac[4] ;//マンハッタン距離の格納場所
main() {
    while (!(memcmp(s, g, sizeof s) == 0))
    {
        getchar();
        for (i = 0; i <= 3; i++) {
            for (n = 0; n <= 3; n++) {
                if (s[i][n - 1] == !NULL) {
                    memcpy(stac, s, sizeof s);
                    stac[i][n] = s[i][n - 1];
                    stac[i][n - 1] = s[i][n];
                    memcpy(stac1, stac, sizeof stac);
                    manstac[0] = ma();
                    print();
                    printf("マンハッタン=%d¥n", manstac[0]);

                }
                if (s[i][n + 1] == !NULL) {
                    memcpy(stac, s, sizeof s);
                    stac[i][n] = s[i][n + 1];
                    stac[i][n + 1] = s[i][n];
                    memcpy(stac2, stac, sizeof stac);
                    manstac[1] = ma();

```

```

        print();
        printf("マンハッタ ン=%d¥n", manstac[1]);

    }
    if (s[i - 1][n] == !NULL) {
        memcpy(stac, s, sizeof s);
        stac[i][n] = s[i - 1][n];
        stac[i - 1][n] = s[i][n];
        memcpy(stac3, stac, sizeof stac);
        manstac[2] = ma();
        print();
        printf("マンハッタ ン=%d¥n", manstac[2]);

    }
    if (s[i + 1][n] == !NULL) {
        memcpy(stac, s, sizeof s);
        stac[i][n] = s[i + 1][n];
        stac[i + 1][n] = s[i][n];
        memcpy(stac4, stac, sizeof stac);
        manstac[3] = ma();
        print();
        printf("マンハッタ ン=%d¥n", manstac[3]);

    }
    for (i = 0; i <= 3; i++) { //最小値を求める
        if (!(0 == manstac[i])) {
            mini = min(mini, manstac[i]);
        }
        printf("¥n %d", manstac[i]);
    }
    printf("¥n mini %d", mini);
    while (!(mini == manstac[nb])) //最小値が格納されてい
る場所を調べる

    {

        nb++;

    }

```

```
        if (nb == 0) {
            memcpy(s, stac1, sizeof stac1);
            s[i][n] = stac1[i][n];
        }
    }
    printf("¥n A:%d", 1);
}
if (nb == 1) {
    memcpy(s, stac2, sizeof stac2);
    printf("¥n A:%d¥n", 2);
}
if (nb == 2) {
    memcpy(s, stac3, sizeof stac3);

    }
}

if (nb == 3) {
    memcpy(s, stac3, sizeof stac3);
    printf("¥n A:%d¥n", 2);

    }

}

}
```

```
int ma(){ //マンハッタン距離
```

```

        int m=0;
        for (i = 0; i < 3; i++){
            for (n = 0; n < 3; n++) {
                m=m + loop(i, n);

            }
        }

        return m;
    }

int print(){//表示
    for (i = 0; i < 3; i++){
        for (n = 0; n < 3; n++){
            printf("%d", stac[i][n]);

        }
        printf("\n");
    }

    return 0;
}

int loop(int i ,int n) {//ループ
    int md=0;
    for (i2 = 0; i2 < 3; i2++) {
        for (n2 = 0; n2 < 3; n2++) {
            if (stac[i][n] == g[i2][n2]) {
                md=md + (n2 - n) + (i2 - i);
                md=abs(md);
            }
        }
    }

    return md;
}

```



### 3 実行結果

```
185
204
837
マンハッタン=10
815
204
837
マンハッタン=12
0
10
12
0
mini 10
A:1
A:2
0
10
12
0
mini 10
0
10
12
0
mini 10
0
10
12
0
mini 10
0
10
12
0
mini 10
A:1
A:2
```

図 4 現在できているところまでの実行結果

- 現状一回目の初期状態からの移行ができているが2回目以降に配列スタート位置の配列を上書きできていないため次に進まない状態になってしまっていると思われる。

### 4 発見的関数推移グラフ

- 結果が出ていないので作成できませんでした。