8 PUZZLE

戴雅萱 劉律琪 游佳霖

報告流程

分工
題目說明
解題方法
程式碼
Running time 比較
數據分析
應用

主要分工

- 戴雅萱 100703006 Data analysis
- 劉律琪 100703007 Coding
- 游佳霖 100703016 PPT
- 劉律琪 戴雅萱 游佳霖 Presentation

UVa 652 - Eight

Sample Input

```
1
2 3 4 1 5 x 7 6 8
```

Input

The first line of the input is an integer N, then a blank line followed by N datasets. There is a blank line between datasets.
In each dataset, you will receive a description of a configuration of the 8 puzzle. The description is just a list of the tiles in their initial positions, with the rows listed from top to bottom, and the tiles listed from left to right within a row, where the tiles are represented by numbers 1 to 8, plus `x'.
1 2 3
x 4 6
7 5 8
is described by this list:
1 2 3 x 4 6 7 5 8

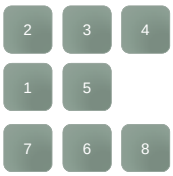
Sample Output

```
ulddrurdllurdruldr
```

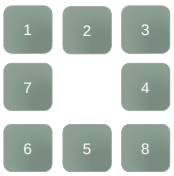
Output

For each dataset, you will print to standard output either the word ``unsolvable'', if the puzzle has no solution, or a string consisting entirely of the letters `r', `l', `u' and `d' that describes a series of moves that produce a solution. The string should include no spaces and start at the beginning of the line. Print a blank line between datasets.

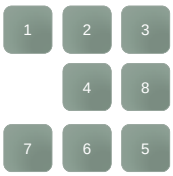
Demo



Demo



Demo



Demo

解題方式

- 解題方式一　ＢＦＳ　　• ＴＬＥ
- 解題方式二　ＢＦＳ優化　• ＡＣ
- 解題方式三　ＩＤＡ＊　　• ＡＣ

前情提要

- 曼哈頓距離：

距離目標位置的距離和

在平面上，座標（x1, y1）的點P1與座標（x2, y2）的點P2的曼哈頓距離為： $|x_1 - x_2| + |y_1 - y_2|$.

解題方法

解題方式一

ＢＦＳ

Code

BFS

TLE版

```
const int D = 3;

struct unit{
    int board[3][3];
    int zx,zy,step,from,dir;
};

int transform(unit &A){
    int result = 0;
    for(int i =0;i < 3;i++){
        for(int j = 0;j < 3;j++){
            result = result*10 + A.board[i][j];
            if(A.board[i][j]==0){
                A.zx = i;
                A.zy = j;
            }
        }
    }
    return result;
}
```

Code

BFS
TLE版

```
std::vector<unit>V;
int BFS(unit start){
    V.clear();
    std::map<int,bool>flag;
    flag[transform(start)] = true;
    start.step = 0;
    V.push_back(start);
    if(transform(start)==123456780) return start.step;
    int dx[4]={1,-1,0,0}, dy[4]={0,0,1,-1};
    for(int i =0;i < V.size();i++){
        unit tmp = V[i];
        for(int j=0;j<4;j++){
            int a =tmp.zx+dx[j];
            int b =tmp.zy+dy[j];
            if(a<0||a>D-1||b<0||b>D-1) continue;
            unit move = tmp;
            move.board[tmp.zx][tmp.zy] = tmp.board[a][b];
            move.board[a][b] = 0;
            if(flag[transform(move)]) continue;
            flag[transform(move)] = true;
            move.step++;
            move.from = i;
            move.dir = j;
            V.push_back(move);
            if(transform(move)==123456780) return move.step;
        }
    }
    return -1;
}
```

BFS

Code

BFS
TLE版

```
void trace(int now){
    if(now==0) return;
    trace(V[now].from);
    switch(V[now].dir){
        case 0:
            printf("d");
            break;
        case 1:
            printf("u");
            break;
        case 2:
            printf("r");
            break;
        case 3:
            printf("l");
            break;
    }
}
```

Code

BFS
TLE版

```
int main(){
    int num;
    unit start;
    scanf("%d",&num);
    while(num!=0){
        for(int i =0;i < D; i++){
            for(int j = 0; j < D; j++ ){
                char s[2];
                scanf("%s",s);
                if(s[0]=='x')
                    start.board[i][j] = 0;
                else
                    start.board[i][j] = atoi(s);
            }
        }
        int step = BFS(start);
        if(step == -1)
            printf("unsolvable\n");
        else{
            trace(V.size()-1);
            printf("\n");
        }
        num--;
    }
    return 0;
}
```

輸入測資

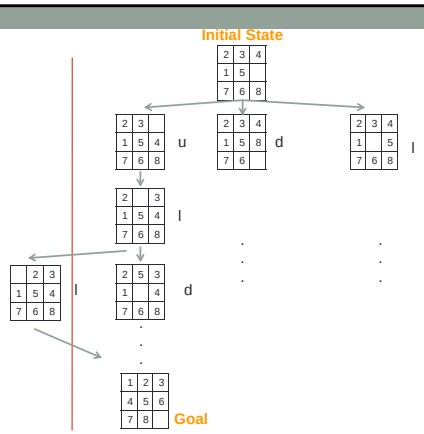
```
[s10007@ghost]~/eight# cat test
6

2 3 4 1 5 x 7 6 8

1 2 3 4 5 x 7 8 6

1 3 2 4 6 5 x 7 8

4 5 6 7 8 x 1 2 3

1 2 3 x 4 6 7 5 8

5 6 7 4 3 2 x 1 8
```

運算時間

```
[s10007@ghost]~/eight# g++ -O2 eightOriginal.cc
[s10007@ghost]~/eight# time ./a.out < test
dlurulldrurdlurdr
d
uurdrulldrurulddr
unsolvable
rdr
uurrdldluurdlurrdldlurdr

real    0m0.428s
user    0m0.384s
sys     0m0.034s
```

Code

BFS
TLE版



解題方法

解題方式二
ＢＦＳ 優化

Code

BFS
AC版

```
const int D = 3;
int power[11];
struct unit{
int zx,zy,step,from,dir;
int trans;
int val(int a,int b){
return trans/power[(2-a)*3+(2-b)]%10;
}
};
const int M=3241993;
std::vector<int>H[M];
bool flag(int x)
{
int y=x%M;
for( int i=0;i<H[y].size();i++)
if(H[y][i]==x)
return true;
H[y].push_back(x);
return false;
}
```

Hash

Code

BFS

AC版

```
std::vector<unit>V;
int BFS(unit start){
flag(start.trans);
start.step = 0;
V.push_back(start);
int dx[4]={1,-1,0,0}, dy[4]={0,0,1,-1};
for(int i =0;i < V.size();i++){
unit tmp = V[i];
int x=tmp.zx,y=tmp.zy;
for(int j=0;j<4;j++){
int a =tmp.zx+dx[j];
int b =tmp.zy+dy[j];
if(a<0||a>D-1||b<0||b>D-1) continue;
unit move = tmp;
move.zx=a;move.zy=b;
move.trans = tmp.trans - move.val(a,b)*power[(2-a)*3+(2-b)] +
move.val(a,b)*power[(2-x)*3+(2-y)];
if(flag(move.trans))continue;
move.step++;
move.from = i;
move.dir = j;
V.push_back(move);
}
}
return -1;
}
```

BFS

Code

BFS

AC版

```
void trace(int now){
if(now==0) return;
switch(V[now].dir){
case 0:
printf("u");
break;
case 1:
printf("d");
break;
case 2:
printf("l");
break;
case 3:
printf("r");
break;
}
trace(V[now].from);
}
```

Code

BFS

AC版

```
int main(){
clock_t cl=clock();
int num;
power[0]=1;
for(int i=1;i<=9;i++)
power[i]=power[i-1]*10;
unit start;
start.trans=123456780;
start.zx=start.zy=2;
BFS(start);
scanf("%d",&num);
int board[D][D];
while(num!=0){
for(int i =0;i < D; i++){
for(int j = 0; j < D; j++ ){
char s[2];
scanf("%s",s);
if(s[0]=='x'||s[0]=='0')
board[i][j] = 0;
else
board[i][j] = atoi(s);
}
}
```

輸入測資

```
[s10007@ghost]~/eight# cat test
6

2 3 4 1 5 x 7 6 8

1 2 3 4 5 x 7 8 6

1 3 2 4 6 5 x 7 8

4 5 6 7 8 x 1 2 3

1 2 3 x 4 6 7 5 8

5 6 7 4 3 2 x 1 8
```

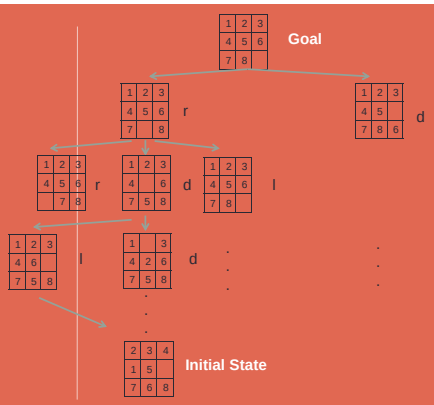
運算時間

```
[s10007@ghost]~/eight# g++ -O2 ACcode.cc
[s10007@ghost]~/eight# time ./a.out < test
ulldrurdllurrdlurd

d

uurrdluldrruuldrd

unsolvable

rdr

uurrdldluruldrdlururdldr

real    0m0.275s
user    0m0.195s
sys     0m0.061s
```

Code

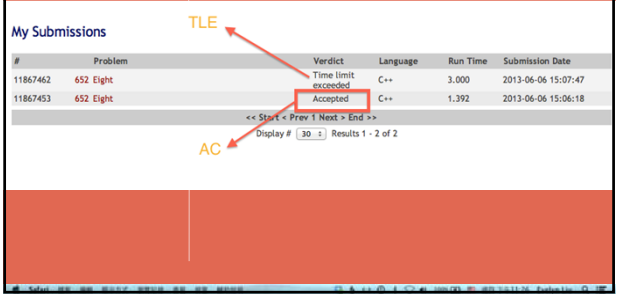
BFS

AC版



Code

BFS



解題方法

解題方式三

ＩＤＡ＊

Code

IDA*

```
const int D=3;
char dir[4] = {'d','r','l','u'};
char path[30];
int pathIndex=0;
int board[D][D];
int ans,dx[]={1,0,0,-1},dy[]={0,1,-1,0};
bool find;
inline void swap(int &a,int &b){int tmp=a;a=b;b=tmp;}
inline int manhattan_dis(int x,int i,int j){
    return abs((x-1)/D-i)+abs((x-1)%D-j);
}
int total_dis(){
    int ans=0;
    for(int i=0;i<D;i++)
        for(int j=0;j<D;j++)
            if(board[i][j]!=0)
                ans+=manhattan_dis(board[i][j],i,j);
    return ans;
}
```

Code

IDA*

```
void DFS(int x,int y,int now_step,int last,int value){
    if(value==0){
        pathIndex=0;
        find=true;
        path[pathIndex++]=dir[3-last];
        return;
    }
    for(int k=0;k<4;k++){
        if(k==last)continue;
        int a=x+dx[k],b=y+dy[k];
        if(a<0||a>=D||b<0||b>=D)continue;

        int new_value=value-
manhattan_dis(board[a][b],a,b)+manhattan_dis(board[a][b],x,y);
        if(new_value+now_step+1<=ans){
            swap(board[x][y],board[a][b]);
            DFS(a,b,now_step+1,3-k,new_value);
            swap(board[x][y],board[a][b]);
          if(find){
          path[pathIndex++]=dir[3-last];
          return;
                                }
                        }
          }
}
```

Code

IDA*

```
void find_zero(int &x,int &y){
    for(int i=0;i<D;i++)
        for(int j=0;j<D;j++)
            if(board[i][j]==0)
            {x=i;y=j;return;}
}
bool unsolvable(int x,int y){
    int cnt=0,a[D][D];
    for(int i=0;i<D;i++)
        for(int j=0;j<D;j++)
            a[i][j]=board[i][j];
    while(x!=D-1||y!=D-1){
        if(x!=D-1)swap(a[x][y],a[x+1][y]),x++;
        if(y!=D-1)swap(a[x][y],a[x][y+1]),y++;
    }
    for(int k=1;k<D*D;k++)
        for(int i=0;i<D;i++)
            for(int j=0;j<D;j++)
                if(a[i][j]==k&&(i!=(k-1)/D||j!=(k-1)%D)){
                    swap(a[i][j],a[(k-1)/D][(k-1)%D]);
                    cnt++;
                }
    return cnt&1;
}
```

Code

IDA*

```
int main(){
    int T;
    scanf("%d",&T);
    while(T--){
        for(int i=0;i<D;i++){
            for(int j=0;j<D;j++){
            char s[2];
            scanf("%s",s);
          if(s[0]=='x')
            board[i][j] = 0;
          else
             board[i][j] = atoi(s);
            }
        }
      int x,y;
      find_zero(x,y);
        if(unsolvable(x
        printf("This puz
not solvable.\n\n");
        else {
        int val=total_di
        find=false;
```

輸入測資

```
[s10007@ghost]~/eight# cat test
6

2 3 4 1 5 x 7 6 8

1 2 3 4 5 x 7 8 6

1 3 2 4 6 5 x 7 8

4 5 6 7 8 x 1 2 3

1 2 3 x 4 6 7 5 8

5 6 7 4 3 2 x 1 8
```

運算時間

```
[s10007@ghost]~/eight# time ./a.out < test
dlurulldrurdllurdr

d

ururdllurrdlulddrr

This puzzle is not solvable.

rdr

urdruuldrdlulurdruldlrdrr

real    0m0.005s
user    0m0.002s
sys     0m0.003s
```

Summary

- Running Time:
  BFS > BFS優化 > IDA*
- Reason:
  1. BFS > BFS優化
     BFS(unit start)的時間複雜度為O(9!/2) (有9!種盤面，其中只有一半是solvable的)，在沒有優化前，走到最終盤面需要嘗試多少盤面就要做多少次BFS；而在優化後總共只需做一次BFS用終點棋盤當起點紀錄完所有可能盤面，因次BFS > BFS優化。
  2. IDA*為最快解
     此法利用DFS，並且用限制步數的方式，讓DFS能節省時間不去找不可能的路徑，又因為使用曼哈頓距離和作為限制步數估計初值，因而達到爆搜剪枝的效果。

延伸及應用

- 神魔之塔

延伸及應用

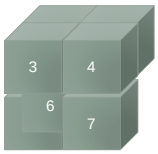
- 立體停車場

延伸及應用

Uva --- 716 Commedia dell' arte

Input

2
2
1 2 3 4
5 7 6 0
2
2 1 3 5
4 6 0 7



Output

Puzzle is unsolvable.
Puzzle can be solved.

謝謝聆聽

THE END