Biên soạn Nguyễn Xuân Huy

S U D O K U

Thân mến tặng các sinh viên
Học viện Bưu chính Viễn Thông

Ngày 15 Tháng 10 Năm2021
Tác giả

Nguyễn Xuân Huy

Sudoku

Sudoku là trò chơi Nhật Bản. Đó là một bảng 9× 9 ô trong đó có một số ô đã chứa sẵn một số số trong khoảng 1 đến 9. Người chơi cần điền nốt các ô còn lại hiện chứa số 0. Kết quả cuối cùng phải là bảng số đáp ứng được các yêu cầu sau đây:

Mọi dòng, mọi cột và mọi khối con 3×3 phải chứa đầy đủ các số từ 1 đến 9.

4	0	0	0	0	0	**6**	0	0
0	**9**	0	0	**8**	0	0	**5**	0
5	0	0	0	0	**9**	**2**	0	0
6	0	0	0	0	0	**1**	0	0
0	**2**	0	0	7	0	0	**3**	0
0	0	**4**	0	0	0	0	0	**9**
0	0	**1**	**3**	0	0	0	0	**8**
0	**3**	0	0	**2**	0	0	**4**	0
0	0	**9**	0	0	0	0	0	**6**

Đề bài

4	1	8	2	5	3	**6**	9	7
3	**9**	2	6	**8**	7	4	**5**	1
5	6	7	4	1	**9**	**2**	8	3
6	8	3	5	9	4	**1**	7	2
9	**2**	5	1	7	6	8	**3**	4
1	7	**4**	8	3	2	5	6	**9**
7	4	**1**	**3**	6	5	9	2	**8**
8	**3**	6	9	**2**	1	7	**4**	5
2	5	**9**	7	4	8	3	1	**6**

Đáp án

Input text file sudoku.inp

```
4 0 0 0 0 0 6 0 0
0 9 0 0 8 0 0 5 0
5 0 0 0 0 9 2 0 0
6 0 0 0 0 0 1 0 0
0 2 0 0 7 0 0 3 0
0 0 4 0 0 0 0 0 9
0 0 1 3 0 0 0 0 8
0 3 0 0 2 0 0 4 0
0 0 9 0 0 0 0 0 6
```

Thuật toán: Quay lui.

Phương án 1. Sau khi đọc dữ liệu vào mảng 9×9 a ta lần lượt duyệt các ô trống (lúc đầu chứa số 0). Với mỗi ô trống a[si][sj] ta gọi hàm Find để chỉnh lại giá trị cần điền cho ô đó. Nếu giá trị hiện hành của ô trống này là v thì ta duyệt từ v+1 đến 9 để tìm giá trị đầu tiên c thỏa điều kiện:

Dòng si không chứa c;
Cột sj không chứa c;
Khối chứa ô (si,sj) không chứa c.

Nếu tìm được giá trị c như trên, ta điền c vào ô a[si][sj] và chuyển qua xử lí ô trống tiếp theo.

Nếu không tìm được giá trị c ta lùi lại ô trống trước đó.

Thuật toán kết thúc thành công khi mọi ô trống đều được điền giá trị hợp lí.

Thuật toán kết thúc vô nghiệm nếu ta quay lui về điểm xuất phát sau khi đã duyệt hết mọi khả năng.

Để xác định được ô trống ta cần lưu lại giá trị input vào mảng 2 chiều s. Mảng hai chiều thứ hai là a sẽ chứa kết quả. Hai biến si và sj dùng để ghi nhận chỉ số dòng và cột của ô hiện hành.

```
int BT1(){
  int i,j;
  DiemXuatPhat();
  cout << "\n Diem xuat phat: si = " << si << "   sj = " << sj;
  while(1){
     if (si > 9) return 1;
     if (si < 1) return 0;
     a[si][sj] = Find();
     if (a[si][sj] > 0) NextCell(); // Tien
     else PredCell(); // Lui
  }

  return 1;
}
```

Thủ tục DiemXuatPhat() tìm ô trống đầu tiên (si,sj) để làm ô xuất phát cho quá trình duyệt.

Hàm Find() trước hết đánh dấu các số đã ghi trên dòng si, cột sj và khối 3×3 chứa ô đang xét (si,sj). Mảng c dùng để đánh dấu được ghi nhận như sau: c[i] = 0 cho biết số i chưa xuất hiện, ngược lại, c[i] > 0 cho biết số i đã xuất hiện trên dòng si hoặc cột sj hoặc trong khối chứa ô (si,sj).

```
int Find(){
  memset(c,0,sizeof(c));
  int i, j, di, dj;
 // Danh dau cac so tren dong si, cot sj
  for (i = 1; i <= 9; ++i){
    c[a[si][i]] = c[a[i][sj]] = 1;
  }
  // Dau khoi: (di,dj)
  di = (si < 4) ? 1 : ((si < 7) ? 4 : 7);
  dj = (sj < 4) ? 1 : ((sj < 7) ? 4 : 7);
  for (i = 0; i < 3; ++i)
    for (j = 0; j < 3; ++j)
        c[a[di+i][dj+j]] = 1;
  // Tim so de cap nhat o (si,sj)
  for (i = a[si][sj]+1; i <= 9; ++i)
    if (c[i] == 0) return i;
  return 0;
}
```

Hai hàm NextCell tìm ô trống sát sau, hàm PredCell tìm ô trống sát trước ô trống đã duyệt.

```
// Tim o trong sau o [si,sj] trong ma tran s
void NextCell(){
   int i, j;
   // Duyet dong si
   for (j = sj+1; j <= 9; ++j)
     if (s[si][j] == 0) {
       sj = j; return;
   }
   // Duyet cac dong tu si+1 .. 9
   for (i = si+1; i <= 9; ++i)
     for (j = 1; j <= 9; ++j)
```

```
        if (s[i][j] == 0){
           si = i; sj = j; return;
        }
   si = sj = 10;
}

// Tim o trong truoc o [si,sj] trong ma tran s
void PredCell(){
   int i, j;
   // Duyet dong si
   for (j = sj-1; j > 0; --j)
     if (s[si][j] == 0) {
       sj = j; return;
   }
   // Duyet cac dong tu si+1 .. 9
   for (i = si-1; i > 0; --i)
     for (j = 9; j > 0; --j)
       if (s[i][j] == 0){
          si = i; sj = j; return;
       }
   si = sj = 0;
}
```

Phương án 2. Ta sử dụng thêm một số mảng hai chiều để ghi nhận các ô trống sát sau và sát trước mỗi ô trống. Trước hết ta duyệt xuôi các ô trống, từ ô trống đầu tiên đến ô trống cuối cùng để thiết lập trị cho các mảng iNext và jNext. iNext[i][j] chứa số hiệu dòng, jNext[i][j] chứa số hiệu cột của ô trống sát trước ô trống (i,j). Sau đó ta duyệt ngược các ô trống để thiết lập trị cho các mảng iPred và jPred. iPrredt[i][j] chứa số hiệu dòng, jPred[i][j] chứa số hiệu cột của ô trống sát sau ô trống (i,j).

```
// To chuc cac tro truoc va sau cho moi a[i][j]
 void Init(){
   int q, ii, jj, i, j;
   // Duyet xuoi
    q = ii = jj = 0;
   for (i = 1; i <= 9; ++i)
     for (j = 1; j <= 9; ++j)
       switch(q){
         case 0: // duyet doan 0
                if (a[i][j] == 0) {
                    iPred[i][j] = ii; jPred[i][j] = jj;
                    ii = i; jj = j;
                 } else q = 1;
              break;
         case 1: // Duyet doan > 0
              if (a[i][j] == 0) {
                 iPred[i][j] = ii; jPred[i][j] = jj;
                 ii = i; jj = j; q = 0;
              }
              break;
       } // switch
       // Duyet nguoc
       q = 0; ii = jj = 10;
       for (i = 9; i > 0; --i)
         for (j = 9; j > 0; --j)
```

```
        switch(q){
          case 0: // duyet doan 0
                 if (a[i][j] == 0) {
                     iNext[i][j] = ii; jNext[i][j] = jj;
                     ii = i; jj = j;
                  } else q = 1;
               break;
          case 1: // Duyet doan > 0
               if (a[i][j] == 0) {
                  iNext[i][j] = ii; jNext[i][j] = jj;
                  ii = i; jj = j; q = 0;
               }
               break;
        } // switch
 }
```

Sau khi thiết lập được các ô trống sát sau và sát trước cho mỗi ô trống ta tổ chức phương án BT2 như sau:

```
int BT2(){
  int i,j;
  DiemXuatPhat();
  cout << "\n Diem xuat phat: si = " << si << "   sj = " << sj;
  while(1){
     if (si > 9) return 1;
     if (si < 1) return 0;
     a[si][sj] = Find();
     i = si; j = sj;
     if (a[si][sj] > 0) { // Tien
       si = iNext[i][j]; sj = jNext[i][j];
     }
     else { // Lui
       si = iPred[i][j]; sj = jPred[i][j];
     }
  }
}
```

Cuối cùng ta viết hàm Test để kiểm tra kết quả có thỏa các điều kiện sudoku hay không.

Chương trình C++

```
/*-----------------------------------------------------
 SUDOKU.CPP
 Phuong an 1. Quay lui binh thuong.
 Phuong an 2. TO CHUC CON TRO TRUOC VA SAU CHO MOI O.
 -----------------------------------------------------*/
#include <iostream>
#include <fstream>
using namespace std;

const int MN = 10;
int a[MN][MN]; // sudoku table
int s[MN][MN];
int iNext[MN][MN], jNext[MN][MN], iPred[MN][MN],jPred[MN][MN];
int c[MN];
```

```
int si,sj; // CHI SO O DANG DUYET
// Hien thi ma tran 2 chieu a[d..c][d..c]
void Print(int a[][MN], int d = 1, int c =  9){
  int i, j;
  for (i = d; i <= c; ++i){
     cout << endl;
    for (j = d; j <= c; ++j)
      cout << " " << a[i][j];
    }
}
// Hien thi mang 1 chieu a[d..c]
void Print(int a[], int d = 1, int c =  9){
  int i;
  cout << endl;
  for (i = d; i <= c; ++i)
     cout << " " << a[i];
}

void Read(){
  int i, j;
  ifstream f("sudoku.inp");
  for (i = 1; i <= 9; ++i)
    for (j = 1; j <= 9; ++j)
      f >> a[i][j];
  f.close();
}

// To chuc cac tro truoc va sau cho moi a[i][j]
 void Init(){
   int q, ii, jj, i, j;
   // Duyet xuoi
    q = ii = jj = 0;
   for (i = 1; i <= 9; ++i)
     for (j = 1; j <= 9; ++j)
       switch(q){
         case 0: // duyet doan 0
                if (a[i][j] == 0) {
                    iPred[i][j] = ii; jPred[i][j] = jj;
                    ii = i; jj = j;
                  } else q = 1;
              break;
         case 1: // Duyet doan > 0
              if (a[i][j] == 0) {
                 iPred[i][j] = ii; jPred[i][j] = jj;
                 ii = i; jj = j; q = 0;
              }
              break;
       } // switch
       // Duyet nguoc
       q = 0; ii = jj = 10;
       for (i = 9; i > 0; --i)
         for (j = 9; j > 0; --j)
       switch(q){
         case 0: // duyet doan 0
                if (a[i][j] == 0) {
                    iNext[i][j] = ii; jNext[i][j] = jj;
                    ii = i; jj = j;
```

```
                } else q = 1;
             break;
          case 1: // Duyet doan > 0
             if (a[i][j] == 0) {
                iNext[i][j] = ii; jNext[i][j] = jj;
                ii = i; jj = j; q = 0;
             }
             break;
       } // switch
 }

int Find(){
  memset(c,0,sizeof(c));
  int i, j, di, dj;
  for (i = 1; i <= 9; ++i){
    c[a[si][i]] = c[a[i][sj]] = 1;
  }
  // Dau khoi: (di,dj)
  di = (si < 4) ? 1 : ((si < 7) ? 4 : 7);
  dj = (sj < 4) ? 1 : ((sj < 7) ? 4 : 7);
  for (i = 0; i < 3; ++i)
    for (j = 0; j < 3; ++j)
        c[a[di+i][dj+j]] = 1;
  for (i = a[si][sj]+1; i <= 9; ++i)
    if (c[i] == 0) return i;
  return 0;
}
// Tim o trong (si,sj) dau tien
void DiemXuatPhat(){
  int i, j;
  si = sj = 0;
  for (i = 1; i <= 9; ++i){
    if (si > 0) return;
    for (j = 1; j <= 9; ++j)
      if (a[i][j] == 0) {
        si = i; sj = j; return;
      }
    }
}

int BT2(){
  int i,j;
  DiemXuatPhat();
  cout << "\n Diem xuat phat: si = " << si << "   sj = " << sj;
  while(1){
     if (si > 9) return 1;
     if (si < 1) return 0;
     a[si][sj] = Find();
     i = si; j = sj;
     if (a[si][sj] > 0) {
       si = iNext[i][j]; sj = jNext[i][j];
     }
     else {
       si = iPred[i][j]; sj = jPred[i][j];
     }
  }
}
```

```
int TestRow(int i){
    int j;
    cout << "\n Test row " << i << ": ";
    memset(c, 0, sizeof(c));
    for (j = 1; j <= 9; ++j)
      if (c[a[i][j]] > 0) {
        cout << " Failed!"; return 0;
      } else c[a[i][j]] = 1;
    cout << " Passed. "; return 1;
}

int TestColum(int i){
    int j;
    cout << "\n Test colum " << i << ": ";
    memset(c, 0, sizeof(c));
    for (j = 1; j <= 9; ++j)
      if (c[a[j][i]] > 0) {
        cout << " Failed!"; return 0;
      } else c[a[j][i]] = 1;
    cout << " Passed. "; return 1;
}

int TestBloc(int di, int dj){
    int i, j;
    cout << "\n Test bloc [" << di << "," << dj << "] : ";
    memset(c, 0, sizeof(c));
    for (i = 0; i < 3; ++i)
      for (j = 0; j < 3; ++j)
      if (c[a[di+i][dj+j]] > 0) {
        cout << " Failed!"; return 0;
      } else c[a[di+i][dj+j]] = 1;
    cout << " Passed. ";
}

int Test() {
  int i, j;
  for (i = 1; i <= 9; ++i)  TestRow(i);
  for (i = 1; i <= 9; ++i)  TestColum(i);
  for (i = 1; i < 8; i += 3)
    for (j = 1; j < 8; j += 3)
      TestBloc(i,j);
}

void Run2(){
  int k;
  cout << "\n SUDOKU Phuong an 2: "
       << " Su dung con tro truoc va sau.\n";
  Read();
  Print(a);
  Init();
  if (BT2()) {
     cout << "\n Result: \n";
     Print(a);
     Test();
  } else cout << "\n No solution.";
}
```

```
// Tim o trong sau o [si,sj] trong ma tran s
void NextCell(){
   int i, j;
   // Duyet dong si
   for (j = sj+1; j <= 9; ++j)
     if (s[si][j] == 0) {
       sj = j; return;
   }
   // Duyet cacs dong tu si+1 .. 9
   for (i = si+1; i <= 9; ++i)
     for (j = 1; j <= 9; ++j)
       if (s[i][j] == 0){
          si = i; sj = j; return;
       }
   si = sj = 10;
}

// Tim o trong truoc o [si,sj] trong ma tran s
void PredCell(){
   int i, j;
   // Duyet dong si
   for (j = sj-1; j > 0; --j)
     if (s[si][j] == 0) {
       sj = j; return;
   }
   // Duyet cac dong tu si+1 .. 9
   for (i = si-1; i > 0; --i)
     for (j = 9; j > 0; --j)
       if (s[i][j] == 0){
          si = i; sj = j; return;
       }
   si = sj = 0;
}

int BT1(){
  DiemXuatPhat();
  cout << "\n Diem xuat phat: si = " << si << "   sj = " << sj;
  while(1){
     if (si > 9) return 1;
     if (si < 1) return 0;
     a[si][sj] = Find();
     if (a[si][sj] > 0) NextCell();
     else PredCell();
  }

  return 1;
}


// Khong dung con tro
void Run1(){
  int k;
  cout << "\n SUDOKU Phuong an 1: "
       << " Khong su dung con tro truoc va sau.\n";
  Read();
  Print(a);
```

```
        memcpy(s,a,sizeof(s));
        if (BT1()) {
           cout << "\n Result: \n";
           Print(a);
           Test();
        } else cout << "\n No solution.";
      }

      main(){
       Run1();
       cout << endl; system("pause");
       Run2();
        cout << endl; system("pause");
        return 0;
      }

/*
DU LIEU TEST

4 0 0 0 0 0 6 0 0
0 9 0 0 8 0 0 5 0
5 0 0 0 0 9 2 0 0
6 0 0 0 0 0 1 0 0
0 2 0 0 7 0 0 3 0
0 0 4 0 0 0 0 0 9
0 0 1 3 0 0 0 0 8
0 3 0 0 2 0 0 4 0
0 0 9 0 0 0 0 0 6
Dap an

4 1 8 2 5 3 6 9 7
3 9 2 6 8 7 4 5 1
5 6 7 4 1 9 2 8 3
6 8 3 5 9 4 1 7 2
9 2 5 1 7 6 8 3 4
1 7 4 8 3 2 5 6 9
7 4 1 3 6 5 9 2 8
8 3 6 9 2 1 7 4 5
2 5 9 7 4 8 3 1 6

-------------------------
0 3 8 0 7 0 0 0 0
0 0 0 0 0 0 0 7 0
0 0 0 8 0 5 0 1 0
0 5 9 0 6 0 3 0 0
0 0 0 0 9 0 8 0 0
4 0 0 1 2 0 9 0 0
5 0 0 0 0 0 0 0 0
0 9 0 0 1 0 0 0 0
2 7 0 9 0 0 0 0 0
Dap an
6 3 8 2 7 1 4 9 5
1 4 5 6 3 9 2 7 8
9 2 7 8 4 5 6 1 3
7 5 9 4 6 8 3 2 1
3 1 2 5 9 7 8 6 4
4 8 6 1 2 3 9 5 7
```

```
5 6 1 3 8 2 7 4 9
8 9 4 7 1 6 5 3 2
2 7 3 9 5 4 1 8 6

--------------------
0 0 0 0 0 0 0 7 0
0 0 0 0 7 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 5 0
0 0 0 0 0 7 0 0 0
0 0 7 8 0 0 0 0 4
8 0 0 0 0 0 0 0 0

Dap an
1 2 3 4 5 6 8 7 9
4 5 6 9 7 8 1 2 3
7 8 9 1 2 3 4 6 5
2 1 4 3 6 5 7 9 8
3 6 5 7 8 9 2 4 1
9 7 8 2 1 4 3 5 6
5 3 1 6 4 7 9 8 2
6 9 7 8 3 2 5 1 4
8 4 2 5 9 1 6 3 7

----------------------

*/
```