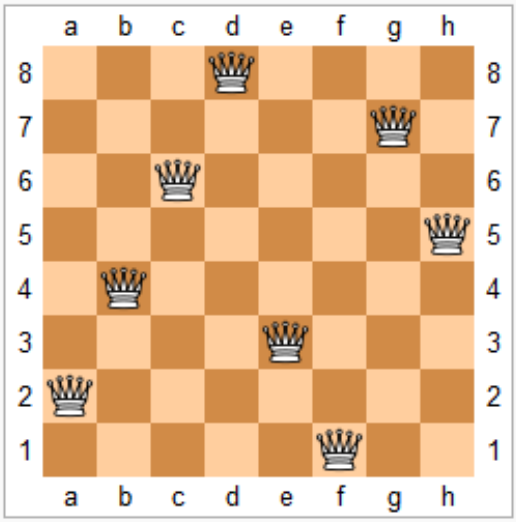
Monday, January 21, 2013

leetcode Question 59: N-Queens

N-Queens

The *n*-queens puzzle is the problem of placing *n* queens on an *n*Ã—*n* chessboard such that no two queens attack each other.



One solution to the eight queens puzzle

Given an integer *n*, return all distinct solutions to the *n*-queens puzzle.

Each solution contains a distinct board configuration of the *n*-queens' placement, where `'Q'` and `'.'` both indicate a queen and an empty space respectively.

For example,
There exist two distinct solutions to the 4-queens puzzle:

```
[
 [".Q..",  // Solution 1
  "...Q",
  "Q...",
  "..Q."],

 ["..Q.",  // Solution 2
  "Q...",
  "...Q",
  ".Q.."]
]
```

Analysis:

The classic recursive problem.

1. Use a int vector to store the current state, A[i]=j refers that the ith row and jth column is placed a queen.
2. Valid state: not in the same column, which is A[i]!=A[current], not in the same diagonal direction: abs(A[i]-A[current]) != r-i

3. Recursion:

Start: placeQueen(0,n)
if current ==n then print result
else
for each place less than n,
place queen
if current state is valid, then place next queen place Queen(cur+1,n)
end for
end if

Source Code:

```
class Solution {
public:

    vector<vector<string> > res;

    void printres(vector<int> A,int n){
        vector<string> r;
        for(int i=0;i<n;i++){
            string str(n,'.');
            str[A[i]]='Q';
            r.push_back(str);
        }
        res.push_back(r);
    }


    bool isValid(vector<int> A, int r){
        for (int i=0;i<r;i++){
            if ( (A[i]==A[r])||(abs(A[i]-A[r])==(r-i))){
                return false;
            }
        }
        return true;
    }

    void nqueens(vector<int> A, int cur, int n){
        if (cur==n){printres(A,n);}
        else{
            for (int i=0;i<n;i++){
                A[cur]=i;
                if (isValid(A,cur)){
                    nqueens(A,cur+1,n);
                }
            }
        }
    }

    vector<vector<string> > solveNQueens(int n) {
        // Start typing your C/C++ solution below
        // DO NOT write int main() function
        res.clear();
        vector<int> A(n,-1);
        nqueens(A,0,n);
```

```
44            return res;
45        }
46    };
```