

N queens (Recursion)

11 April 2022 17:11

<https://nados.io/question/n-queens> -> Question link

```

#include<iostream>
#include<vector>
#include<string>
using namespace std;
bool ispossible(vector<vector<int>> v, int row, int col)
{
    for(int i=row-1,j=col;i>=0;j--)
    {
        if(v[i][j]>0)
        {
            return false;
        }
    }
    for(int i=row-1,j=col-1;i>=0&&j>=0;i--,j--)
    {
        if(v[i][j]>0)
        {
            return false;
        }
    }
    for(int i=row-1,j=col+1;i>=0&&j<v[i].size();i--,j++)
    {
        if(v[i][j]>0)
        {
            return false;
        }
    }
    return true;
}

```

At each coordinates it check, whether it is good to place the queen or not.



```

void queens(vector<vector<int>> v,int n,int row,string asf)
{
    if(row==v.size())
    {
        cout<<asf<<endl;
        return;
    }
    for(int col=0;col<v.size();col++)
    {
        if(ispossible(v,row,col)==true)
        {
            v[row][col]=1;
            //cout<<"d"<<endl;
            queens(v,n,row+1, asf+to_string(row)+" "+to_string(col)+", ");
            v[row][col]=0;
        }
    }
}
int main()
{
    int n;
    cin>>n;

    vector<vector<int>> v(n , vector<int> (n, 0));

    queens(v,n,0,"");
}

```

Queen function -

If row is at last. It means their is a way. So, print that path.

In every column, we placed queen at each rows of that particular column, and check ispossible to placed that queen at that position.

- If "yes" - then change that coordinate to 1. And after returning back in backtracking from last to first, we change the value again to 0. So, that when we trace next new fresh path, their will be no ones.

We used backtracking approach.... That means once we trace a path, we come back by removing all the temporary calculations. Then again trace new path from starting.

