**RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY**

**SESSIONAL TASK-08**

COURSE NAME: SESSIONAL BASED ON CSE-2201

COURSE CODE: CSE-2102

**SUBMITTED TO-**

**BIPRODIP PAL**

Assistant Professor

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

**SUBMITTED BY-**

**SRABONTI DEB**

Roll- 1803163

Section-C

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

SUBMISSION DATE – 30 JULY, 2021

# Problem Statement: Using backtracking, solve the N queens problem. For any N taken as input, your code should find out the goal nodes as well as the bounding nodes (from where no more nodes are checked along that path and backtracking occurred).

**Code:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85 | #include<bits/stdc++.h>  **using** **namespace** std;  **int** arr[**1000**][**1000**],k=**0**,n;  vector <**int**> r;  **void** **n\_queen**()  {  **int** i;  **for**(i=**0**; i<n; i++)  {  **bool** var = false;  r.push\_back(i);  **for**(**int** j=k; j>=**0**; j--)  {  **if**(var)  **break**;  **if**(arr[j][i])  var = true;  }  **for**(**int** j=k,l=i; j>=**0**; j--,l--)  {  **if**(var)  **break**;  **if**(arr[j][l])  var = true;  }  **for**(**int** j=k,l=i; j>=**0**; j--,l++)  {  **if**(var)  **break**;  **if**(arr[j][l])  var = true;  }  **if**(var)  {  cout<<"Backtrack From Node:";  **for**(**int** j=**0**; j<r.size(); j++)  cout<<r[j]+**1**<<" ";  cout<<endl;  r.pop\_back();  **continue**;  }  **if**(r.size() == n)  {  cout<<"Solution: ";  **for**(**int** j=**0**; j<r.size(); j++)  cout<<r[j]+**1**<<" ";  cout<<endl;  r.pop\_back();  **return**;  }  cout<<"Backtrack From Node:";  **for**(**int** j=**0**; j<r.size(); j++)  cout<<r[j]+**1**<<" ";  cout<<endl;  arr[k][i] = **1**;  k++;  n\_queen();  r.pop\_back();  k--;  arr[k][i] = **0**;  }  **return**;  }  **int** **main**()  {  cout<<"Enter Input: ";  cin>>n;  **int** i;  **for**(i=**0**; i<n; i++)  {  r.push\_back(i);  arr[k][i] = **1**;  k++;  n\_queen();  k--;  arr[k][i] = **0**;  r.pop\_back();  }  **return** **0**;  } |

**Output:**

Enter Input: 5

Backtrack From Node:1 1

Backtrack From Node:1 2

Backtrack From Node:1 3

Backtrack From Node:1 3 1

Backtrack From Node:1 3 2

Backtrack From Node:1 3 3

Backtrack From Node:1 3 4

Backtrack From Node:1 3 5

Backtrack From Node:1 3 5 1

Backtrack From Node:1 3 5 2

Backtrack From Node:1 3 5 2 1

Backtrack From Node:1 3 5 2 2

Backtrack From Node:1 3 5 2 3

Solution: 1 3 5 2 4

Backtrack From Node:1 3 5 3

Backtrack From Node:1 3 5 4

Backtrack From Node:1 3 5 5

Backtrack From Node:1 4

Backtrack From Node:1 4 1

Backtrack From Node:1 4 2

Backtrack From Node:1 4 2 1

Backtrack From Node:1 4 2 2

Backtrack From Node:1 4 2 3

Backtrack From Node:1 4 2 4

Backtrack From Node:1 4 2 5

Backtrack From Node:1 4 2 5 1

Backtrack From Node:1 4 2 5 2

Solution: 1 4 2 5 3

Backtrack From Node:1 4 3

Backtrack From Node:1 4 4

Backtrack From Node:1 4 5

Backtrack From Node:1 5

Backtrack From Node:1 5 1

Backtrack From Node:1 5 2

Backtrack From Node:1 5 2 1

Backtrack From Node:1 5 2 2

Backtrack From Node:1 5 2 3

Backtrack From Node:1 5 2 4

Backtrack From Node:1 5 2 5

Backtrack From Node:1 5 3

Backtrack From Node:1 5 4

Backtrack From Node:1 5 5

Backtrack From Node:2 1

Backtrack From Node:2 2

Backtrack From Node:2 3

Backtrack From Node:2 4

Backtrack From Node:2 4 1

Backtrack From Node:2 4 1 1

Backtrack From Node:2 4 1 2

Backtrack From Node:2 4 1 3

Backtrack From Node:2 4 1 3 1

Backtrack From Node:2 4 1 3 2

Backtrack From Node:2 4 1 3 3

Backtrack From Node:2 4 1 3 4

Solution: 2 4 1 3 5

Backtrack From Node:2 4 1 4

Backtrack From Node:2 4 1 5

Backtrack From Node:2 4 2

Backtrack From Node:2 4 3

Backtrack From Node:2 4 4

Backtrack From Node:2 4 5

Backtrack From Node:2 5

Backtrack From Node:2 5 1

Backtrack From Node:2 5 1 1

Backtrack From Node:2 5 1 2

Backtrack From Node:2 5 1 3

Backtrack From Node:2 5 1 4

Backtrack From Node:2 5 1 4 1

Backtrack From Node:2 5 1 4 2

Backtrack From Node:2 5 1 4 3

Backtrack From Node:2 5 1 4 4

Backtrack From Node:2 5 1 4 5

Backtrack From Node:2 5 1 5

Backtrack From Node:2 5 2

Backtrack From Node:2 5 3

Backtrack From Node:2 5 3 1

Backtrack From Node:2 5 3 1 1

Backtrack From Node:2 5 3 1 2

Backtrack From Node:2 5 3 1 3

Solution: 2 5 3 1 4

Backtrack From Node:2 5 3 2

Backtrack From Node:2 5 3 3

Backtrack From Node:2 5 3 4

Backtrack From Node:2 5 3 5

Backtrack From Node:2 5 4

Backtrack From Node:2 5 5

Backtrack From Node:3 1

Backtrack From Node:3 1 1

Backtrack From Node:3 1 2

Backtrack From Node:3 1 3

Backtrack From Node:3 1 4

Backtrack From Node:3 1 4 1

Backtrack From Node:3 1 4 2

Backtrack From Node:3 1 4 2 1

Backtrack From Node:3 1 4 2 2

Backtrack From Node:3 1 4 2 3

Backtrack From Node:3 1 4 2 4

Solution: 3 1 4 2 5

Backtrack From Node:3 1 4 3

Backtrack From Node:3 1 4 4

Backtrack From Node:3 1 4 5

Backtrack From Node:3 1 5

Backtrack From Node:3 2

Backtrack From Node:3 3

Backtrack From Node:3 4

Backtrack From Node:3 5

Backtrack From Node:3 5 1

Backtrack From Node:3 5 2

Backtrack From Node:3 5 2 1

Backtrack From Node:3 5 2 2

Backtrack From Node:3 5 2 3

Backtrack From Node:3 5 2 4

Solution: 3 5 2 4 1

Backtrack From Node:3 5 2 5

Backtrack From Node:3 5 3

Backtrack From Node:3 5 4

Backtrack From Node:3 5 5

Backtrack From Node:4 1

Backtrack From Node:4 1 1

Backtrack From Node:4 1 2

Backtrack From Node:4 1 3

Backtrack From Node:4 1 3 1

Backtrack From Node:4 1 3 2

Backtrack From Node:4 1 3 3

Backtrack From Node:4 1 3 4

Backtrack From Node:4 1 3 5

Backtrack From Node:4 1 3 5 1

Solution: 4 1 3 5 2

Backtrack From Node:4 1 4

Backtrack From Node:4 1 5

Backtrack From Node:4 1 5 1

Backtrack From Node:4 1 5 2

Backtrack From Node:4 1 5 2 1

Backtrack From Node:4 1 5 2 2

Backtrack From Node:4 1 5 2 3

Backtrack From Node:4 1 5 2 4

Backtrack From Node:4 1 5 2 5

Backtrack From Node:4 1 5 3

Backtrack From Node:4 1 5 4

Backtrack From Node:4 1 5 5

Backtrack From Node:4 2

Backtrack From Node:4 2 1

Backtrack From Node:4 2 2

Backtrack From Node:4 2 3

Backtrack From Node:4 2 4

Backtrack From Node:4 2 5

Backtrack From Node:4 2 5 1

Backtrack From Node:4 2 5 2

Backtrack From Node:4 2 5 3

Solution: 4 2 5 3 1

Backtrack From Node:4 2 5 4

Backtrack From Node:4 2 5 5

Backtrack From Node:4 3

Backtrack From Node:4 4

Backtrack From Node:4 5

Backtrack From Node:5 1

Backtrack From Node:5 1 1

Backtrack From Node:5 1 2

Backtrack From Node:5 1 3

Backtrack From Node:5 1 4

Backtrack From Node:5 1 4 1

Backtrack From Node:5 1 4 2

Backtrack From Node:5 1 4 3

Backtrack From Node:5 1 4 4

Backtrack From Node:5 1 4 5

Backtrack From Node:5 1 5

Backtrack From Node:5 2

Backtrack From Node:5 2 1

Backtrack From Node:5 2 2

Backtrack From Node:5 2 3

Backtrack From Node:5 2 4

Backtrack From Node:5 2 4 1

Backtrack From Node:5 2 4 1 1

Backtrack From Node:5 2 4 1 2

Solution: 5 2 4 1 3

Backtrack From Node:5 2 4 2

Backtrack From Node:5 2 4 3

Backtrack From Node:5 2 4 4

Backtrack From Node:5 2 4 5

Backtrack From Node:5 2 5

Backtrack From Node:5 3

Backtrack From Node:5 3 1

Backtrack From Node:5 3 1 1

Backtrack From Node:5 3 1 2

Backtrack From Node:5 3 1 3

Backtrack From Node:5 3 1 4

Backtrack From Node:5 3 1 4 1

Solution: 5 3 1 4 2

Backtrack From Node:5 3 1 5

Backtrack From Node:5 3 2

Backtrack From Node:5 3 3

Backtrack From Node:5 3 4

Backtrack From Node:5 3 5

Backtrack From Node:5 4

Backtrack From Node:5 5

Process returned 0 (0x0) execution time : 7.954 s