

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 #include<bits/stdc++.h>
2 using namespace std;
3
4 int arr[1000][1000], k=0, n;
5 vector <int> r;
6
7 void n_queen()
8 {
9     int i;
10    for(i=0; i<n; i++)
11    {
12        bool var = false;
13        r.push_back(i);
14        for(int j=k; j>=0; j--)
15        {
16            if(var)
17                break;
18            if(arr[j][i])
19                var = true;
20        }
21        for(int j=k, l=i; j>=0; j--, l--)
22        {
23            if(var)
24                break;
25            if(arr[j][l])
26                var = true;
27        }
28        for(int j=k, l=i; j>=0; j--, l++)
29        {
30            if(var)
31                break;
32            if(arr[j][l])
33                var = true;
34        }
35        if(var)
36        {
37            cout<<"Backtrack From Node:";
38            for(int j=0; j<r.size(); j++)
39                cout<<r[j]+1<<" ";
40            cout<<endl;
41            r.pop_back();
42            continue;
43        }
44    }
```

```

45     if(r.size() == n)
46     {
47         cout<<"Solution: ";
48         for(int j=0; j<r.size(); j++)
49             cout<<r[j]+1<<" ";
50         cout<<endl;
51         r.pop_back();
52         return;
53     }
54
55     cout<<"Backtrack From Node:";
56     for(int j=0; j<r.size(); j++)
57         cout<<r[j]+1<<" ";
58     cout<<endl;
59     arr[k][i] = 1;
60     k++;
61     n_queen();
62     r.pop_back();
63     k--;
64     arr[k][i] = 0;
65 }
66 return;
67 }
68
69 int main()
70 {
71     cout<<"Enter Input: ";
72     cin>>n;
73     int i;
74     for(i=0; i<n; i++)
75     {
76         r.push_back(i);
77         arr[k][i] = 1;
78         k++;
79         n_queen();
80         k--;
81         arr[k][i] = 0;
82         r.pop_back();
83     }
84     return 0;
85 }

```

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