

LAB PROGRAMS-9:

→ Longest increasing subsequence:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int max(int a, int b)
```

```
{ return (a > b) ? a : b; }
```

```
int main()
```

```
{ int n, m, i, j;
```

```
printf("Enter the no. of integers: ");
```

```
scanf("%d", &n);
```

```
int a[n], t[n];
```

```
for(i=0; i<n; i++) { t[i]=1; }
```

```
for(i=0; i<n; i++) { scanf("%d", &a[i]); }
```

```
for(i=1; i<n; i++)
```

```
{ for(j=0; j<=i-1; j++)
```

```
{ if(a[j] < a[i])
```

```
{ t[i] = max(t[i], t[j]+1); } } }
```

```
m=0;
```

```
for(i=0; i<n; i++) { if(m < t[i]) { m = t[i]; } }
```

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DAA

Name K.S.I. SIVANI

Roll No. -052

Page No. 113

```
printf("Length of the longest increasing subsequence: %d",  
return 0;
```

```
}
```

O/P:

Enter the no. of integers: 7

3

4

-1

0

6

2

3

Length of the longest increasing subsequence: 4.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DAA

Name K.S.I. SIVANI Roll No. -052 Page No. 114

PRELAB QUESTIONS-10

1) Write the control abstraction for Backtracking design strategy:

def backtracking (problem, solution) :

if is_complete(solution) :

return solution.

candidates = generate_candidates (problem, solution)

for candidate in candidates :

if is_valid(candidate) :

solution.add(candidate)

result = backtracking(problem, solution)

if result is not None :

return result.

solution.remove(candidate).

return None.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DAA

Name K.S.I. SIVANI Roll No. -052 Page No. 115

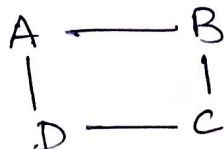
- 2) Write the algorithm to place 'n' queens on a $N \times N$ chess board.

```
int solveQueens(int col)
{
    if (col >= N)
        return 1;
    for (int i = 0; i < N; i++)
    {
        if (isSafe(i, col))
        {
            board[i][col] = 1;
            if (solveQueens(col+1))
                return 1;
            board[i][col] = 0;
        }
    }
}
```

- 3) What is chromatic number? Give an example:

The chromatic number is a concept in graph theory that represents the min. no. of colours needed to colour the vertices of a graph such that no 2 adjacent vertices have same colour.

Ex:



chromatic no = 2.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

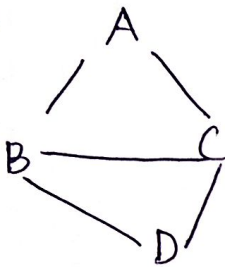
NAME OF THE LABORATORY : DAA

Name K.S.I. SIVANI Roll No. — 052 Page No. 116

4) What is Hamiltonian cycle? Give an example.

A Hamiltonian cycle is a path that visits each vertex exactly once and ends at the starting vertex.

Ex:



$A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$

PRELAB PROGRAMS-10

1) Implement N-queens ~~back~~ problem with back tracking.

```
#include <stdio.h>
```

```
#define N 4
```

```
int board[N][N];
```

```
int isSafe(int row, int col)
```

```
{ int i, j;
```

```
for (i=0; i<col; i++)
```

```
{ if (board[row][i]) return 0; }
```

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF

: CSE

NAME OF THE LABORATORY : DPA

Name K.S.I. SIVANI

Roll No. - 052

Page No. 117

```
for(i=row, j=col; i>=0 && j>=0; i--, j--)  
{ if (board[i][j]) return 0; }
```

```
for(i=row, j=col; j>=0 && i<N; i++, j--)  
{ if (board[i][j]) return 0; }
```

```
return 1; }
```

```
int solveNQueens(int col)
```

```
{ if (col>=N) return 1;
```

```
for(int i=0; i<N; i++)
```

```
{ if (isSafe(i, col))
```

```
{ board[i][col]=1;
```

```
if (solveNQueens(col+1)) return 1;
```

```
board[i][col]=0; } }
```

```
return 0; }
```

```
void print_board()
```

```
{ for(int i=0; i<N; i++)
```

```
{ for(int j=0; j<N; j++)
```

```
{ printf("%c", board[i][j] ? 'Q' : '-'); }
```


VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DAA

Name K.S.I. SIVANI

Roll No. 052

Page No. 118

```
printf("\n"); } }  
int main()  
{ for(int i=0; i<N; i++)  
  { for(int j=0; j<N; j++)  
    { board[i][j]=0; } }  
  if (solveNQueens(0))  
    { printf("Solution:\n");  
      print_board(); }  
  else  
    { printf("No solution exists for the given  
      boardsize\n"); }  
  return 0;  
}
```

2) Implement Graph colouring with back tracking :

```
#include <stdio.h>
```

```
#define V 4.
```

```
int graph[V][V] = { {0, 1, 1, 1}, {1, 0, 1, 0},  
                     {1, 1, 0, 1}, {1, 0, 1, 0} };
```

O/P:

— — — 9 —

9 — — — —

— — — — 9

— 9 — — —

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF

: CSE

NAME OF THE LABORATORY : DAA

Name K.S.I. SIVANI

Roll No. -052

Page No.

119

```
int m=3; int colors[v];
int isSafe(int v, int c)
{
    for(int i=0; i<v; i++)
    {
        if(graph[v][i] && colors[i]==c) return 0;
    }
    return 1;
}

int graphColouringUtil(int v)
{
    if(v==V) return 1;
    for(int c=1; c<=m; c++)
    {
        if(isSafe(v, c))
        {
            colors[v]=c;
            if(graphColouringUtil(v+1)) return 1;
            colors[v]=0;
        }
    }
    return 0;
}
```

```
void printColors()
{
    printf("Vertex\t Colour\n");
    for(int i=0; i<V; i++)
    {
        printf("%d\t %d\n", i, colors[i]);
    }
}
```

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DAA

Name K.S.I. SIVANI Roll No. 1052 Page No. 120

```
int main()
{ for (int i = 0; i < V; i++)
    { colors[i] = 0; }
  if (graphColoringUtil(v))
    { printf("Solution:\n");
      printColors(); }
  else
    { printf("No solution exists for the
             given graph & number of colors.\n");
    }
  return 0; }
```

O/P:

Solution:

Vertex	Color
0	1
1	2
2	3
3	2