Danyl Fernandes
2020012004 (72)
24-04-2021

AOA Experiment 8

Aim:

To implement & analyze Graph Coloring Problem using Backtracking approach:

}

Implementation:

```
#include<stdio.h>
#include<conio.h>
int m, n;
int c=0,count=0;
int G[20][20], x[50];
void nextValue(int k)
{
int j;
while(1)
{
x[k]=(x[k]+1)%(m+1);
if(x[k]==0)
{
return;
}
for(j=1;j<=n;j++)
{
if(G[k][j]==1&&x[k]==x[j])
break;
}
if(j==(n+1))
{
return;
}
}
```

```
}
void MColoring(int k)
{
int i;
while(1)
{
nextValue(k);
if(x[k]==0)
{
return;
}
if(k==n)
{
c=1;
for(i=1;i<=n;i++)
{
printf("%d ", x[i]);
}
count++;
printf("\n");
}
else
MColoring(k+1);
}
}
int main()
{
```

```
int i, j;
int temp;
printf("\nEnter the number of Vertices: " );
scanf("%d", &n);
printf("Enter the needed values:\n");
printf("If edge exists then enter 1 else enter 0 \n");
for(i=1;i<=n;i++)
{
for(j=1;j<=n;j++)
{
printf("G[%d][%d]: ",i,j);
scanf("%d", &G[i][j]);
}
printf("\n");
}
printf("\n__________________________________________\n");
printf("The adjacency matrix: \n");
for(i=1;i<=n;i++)
{
for(j=1;j<=n;j++)
{
printf("%d ",G[i][j]);
}
printf("\n");
}
printf("\n__________________________________________\n");
printf("\nPossible Solutions are\n");
```

```
for(m=1;m<=n;m++)
{
if(c==1)
{
break;
}
MColoring(1);
}
printf("\nThe chromatic number is %d", m-1);
printf("\nThe total number of solutions is %d", count);
getch();
}
```

Output:

```
Enter the number of Vertices: 4
Enter the needed values:
If edge exists then enter 1 else enter 0
G[1][1]: 0
G[1][2]: 1
G[1][3]: 0
G[1][4]: 0

G[2][1]: 1
G[2][2]: 0
G[2][3]: 1
G[2][4]: 1

G[3][1]: 0
G[3][2]: 1
G[3][3]: 0
G[3][4]: 0

G[4][1]: 0
G[4][2]: 1
G[4][3]: 0
G[4][4]: 0
```

```
________________________________________
The adjacency matrix:
0 1 0 0
1 0 1 1
0 1 0 0
0 1 0 0


________________________________________

Possible Solutions are
1 2 1 1
2 1 2 2

The chromatic number is 2
The total number of solutions is 2
```

Danyl Fernandes
2020012004 (72)

Exp 08

Theory :

- The graph colourning is a classical problem in combinatorics. There are several variants of colouring problems.

- It is efficently solved by the backtracking strategy.

Problem Desc.

- Consider (G) = (V, E) be an undirected graph where V is a set of nodes & E is a set of edges.

- Then the graph colouring problem asks to assign m colours to the vertices have a similar colour. It is called the colouring problem.

- If the same problem is applied to Colour edges of a grph G so that any two adjcent edges are coloured with a different colours, then it is known as an edge colouring problem

Algorithm :

M_colouring (k) {

Daryl Fernandes
2020812004 (72)

```
repeat {
    NextColor (k) ;
    if (x[k] = 0)
        return;
    if (k = n)
        write [x [1 : n]) ;
    else MColouring (k+1)
}
until (false) ;
}

NextValue (k) {
  repeat {
    x[k] := (x[k] +1 ) mod (m+1) ;
    if (x[k] = 0) then return ;
    for (j := 1 to n do {
          if ((a[k, j] ≠ 0 & x[k] = x[j]))
              then break ;
    }
    if (j = n+1) then return
  } until (false)
}
```

Analysis :

- Consider a given graph G has n vertices & at most m colours are available to colour it.

- In the state space tree of the decision problem, at level i, the tree has m nodes representing problem states:

Danyl Fernandes
2020012004(72)

- So, the total number of internal nodes in a state of space tree is $\Sigma_0\ 0 \le k \le n-1$ m^k

- Therefore total no of nodes in space tree would be

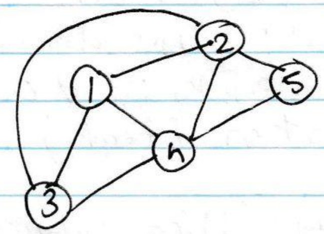
$$1 + M + M^2 + M^3 \cdots + M^n$$

$$T(n) = 1 + M + M^2 + M^3 \cdots + M^n$$

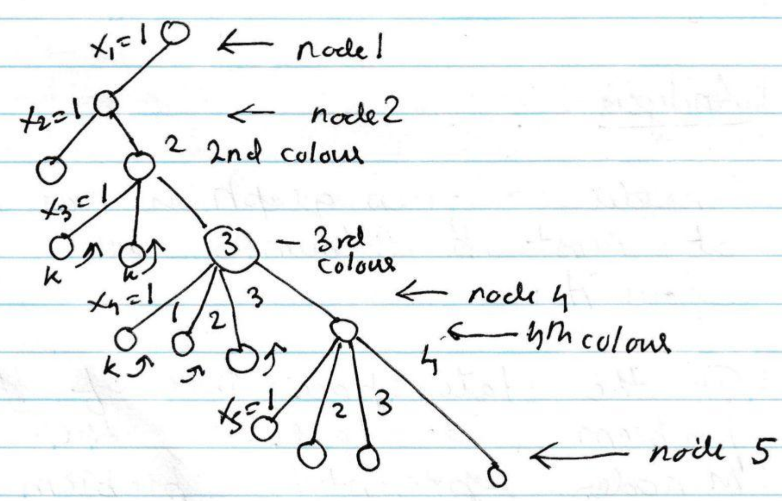
$$= \frac{M^N + 1 - 1}{M - 1}$$

$$\therefore T(n) = O(M^n)$$

<u>Example</u> :

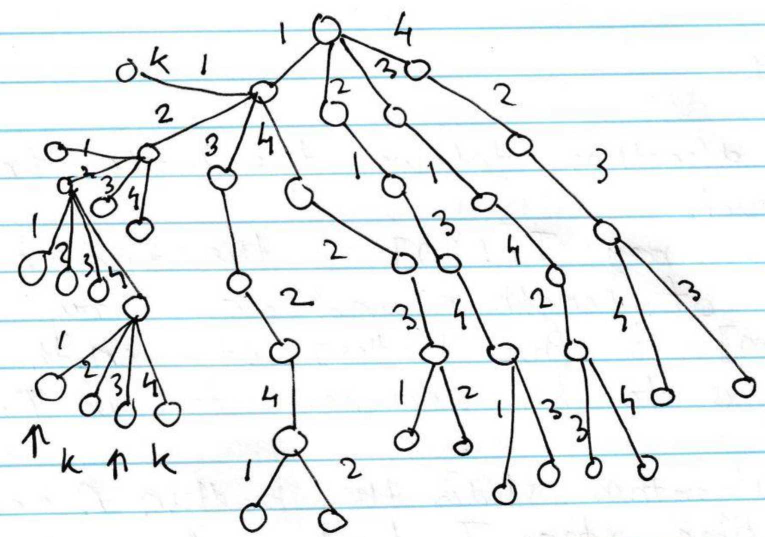


<u>Solution 1</u>



Danyl Fernandes
2020012004 (72)

Solution 2 :



Conclusion : The implementation of m - colouring was successful with the analysis for the same .