ACPCE

Where Knowledge is Second Nature

Jawahar Education Societys Annasaheb Chudaman Patil College of Engineering, Kharghar, Navi Mumbai

NAME: PRIYUSH BHIMRAO KHOBRAGADE

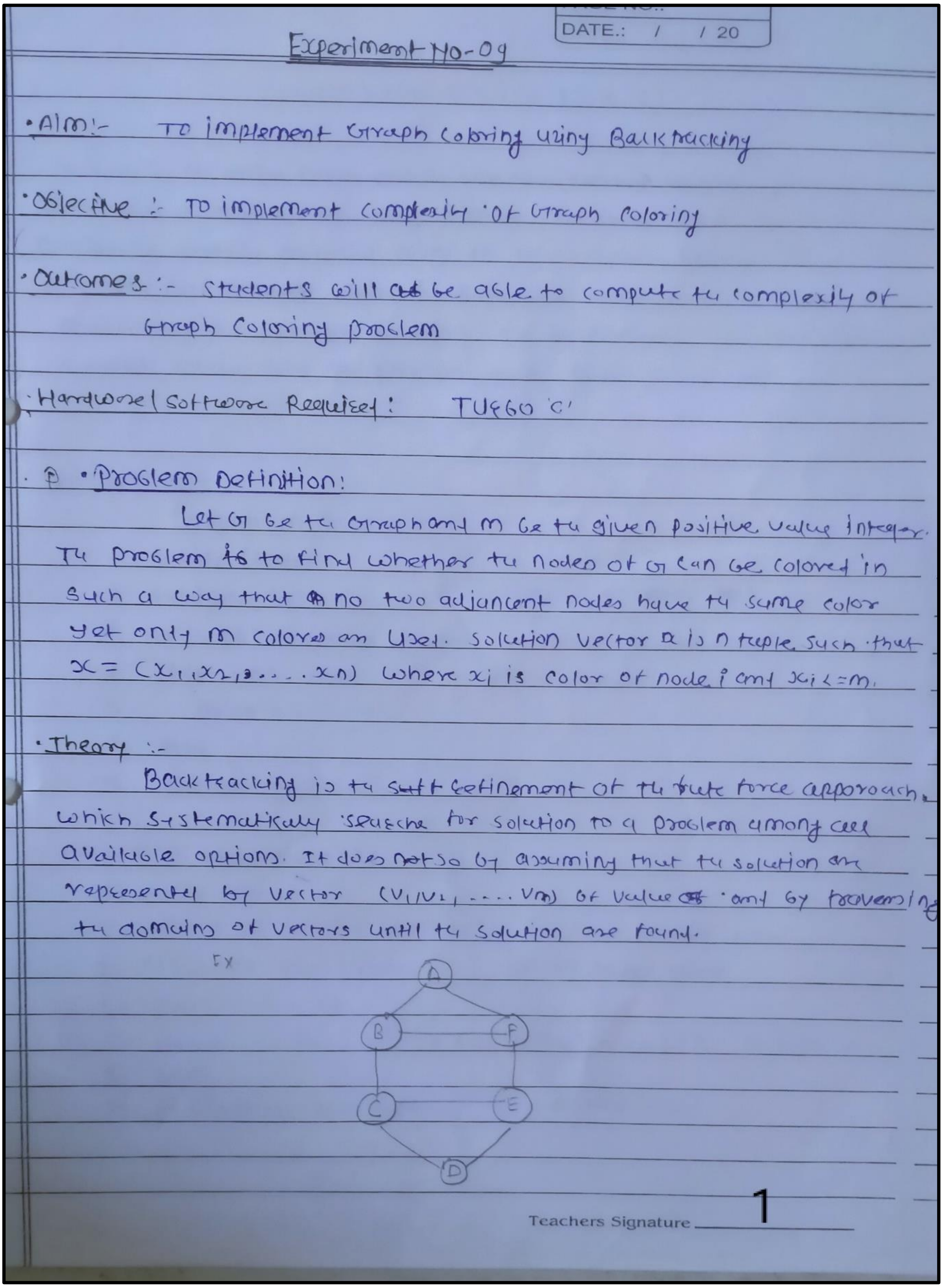
PRN NO: 211112018

Roll No: 52

SUBJECT: Analysis of Algorithms Lab

Implement a C program for Graph Colouring using Backtracking Approach.

EXPERMINT: 09



DATE.: / / 20

Experiment No-09

• Aim:- To implement Graph coloring using Backtracking

• Objective :- To implement complexity of Graph coloring

• Outcomes :- students will ~~not~~ be able to compute the complexity of Graph coloring problem

• Hardware/ Software Required: TURBO 'C'

• ~~P~~ • Problem Definition:

Let G be the Graph and m be the given positive value integer. The problem is to find whether the nodes of G can be colored in such a way that ~~A~~ no two adjuncent nodes have the same color yet only m colors are used. solution vector x is n tuple such that x = (x1, x2, ... xn) where xi is color of node i and xi <= m.

• Theory :-

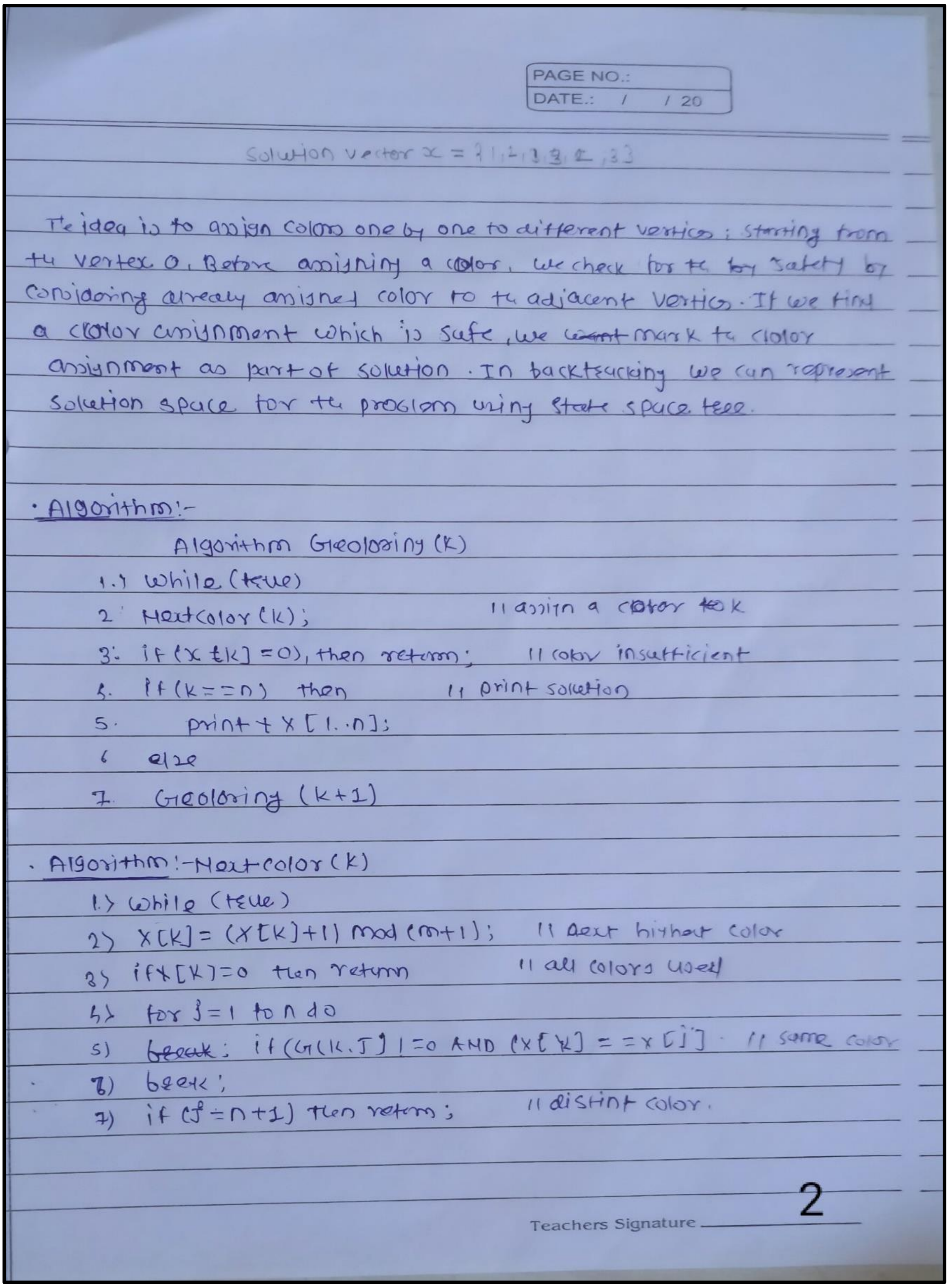
Backtracking is the self refinement of the brute force approach. which systematically searche for solution to a problem among all available options. It does not so by assuming that the solution are represented by vector (v1, v2, ... vm) of value ~~of~~ and by traversing the domains of vectors until the solution are found.

Ex

Teachers Signature 1

PRIYUSH B.KHOBRAGADE

Implement a C program for Graph Colouring using Backtracking Approach.



PAGE NO.:
DATE.: / / 20

Solution vector x = {1,2,3,3,2,3}

Te idea is to assign colors one by one to different vertices; starting from the vertex 0. Before assigning a color, we check for the ~~by~~ safety by considering already assigned color to the adjacent vertices. If we find a color assignment which is safe, we ~~want~~ mark the color assignment as part of solution. In backtracking we can represent solution space for the problem using state space tree.

· Algorithm:-

```
        Algorithm Gcoloring (K)
   1.) While (true)
   2   Nextcolor (k);                    // assign a color to k
   3.  if (x [k] = 0), then return;      // color insufficient
   4.  if (k == n)   then                // print solution
   5.      print x [1..n];
   6   else
   7.  Gcoloring (k+1)
```

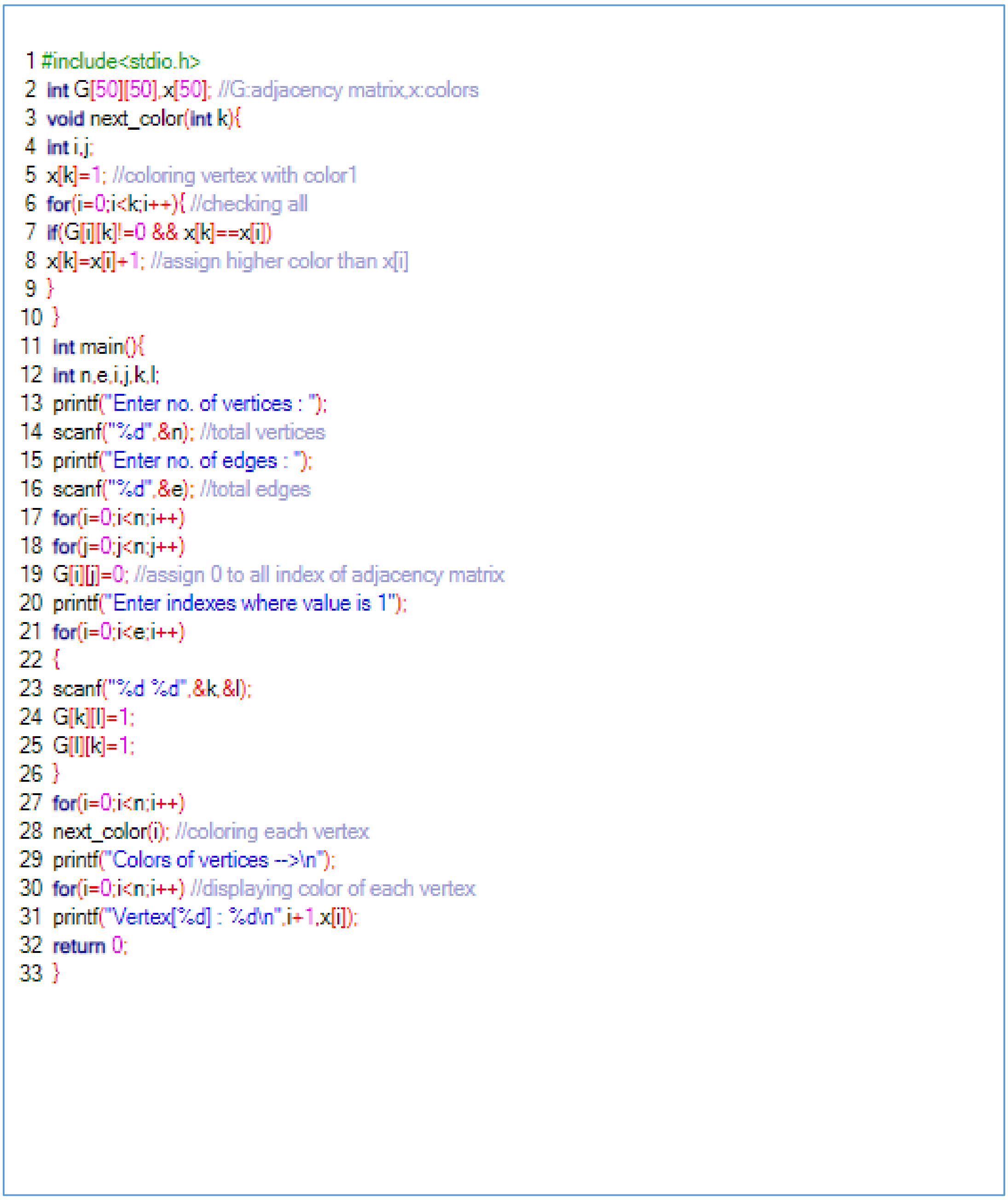
· Algorithm:-Nextcolor (K)

```
   1.> while (true)
   2) x[k] = (x[k]+1) mod (m+1);     // next highest color
   3) if x[k] = 0  then return       // all colors used
   4) for j = 1 to n do
   5) ~~break~~; if (G(k, j] != 0 AND (x[k] == x[j]]   // same color
   6) break;
   7) if (j = n+1) then return;      // distint color.
```

Teachers Signature

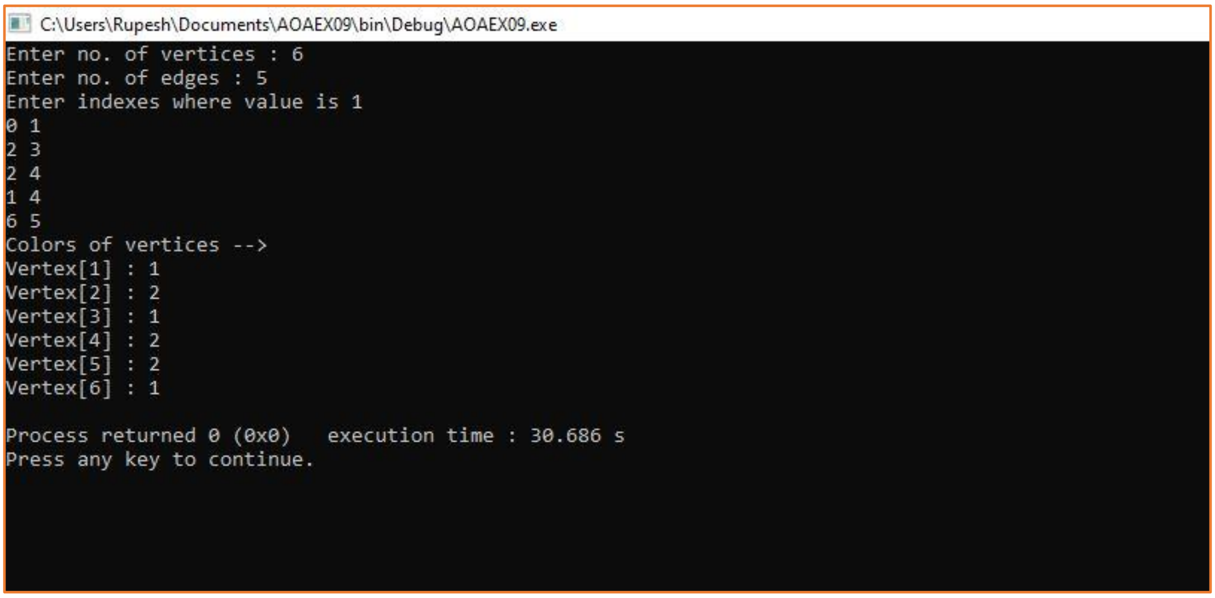
Implement a C program for Graph Colouring using Backtracking Approach.

Input:



```c
#include<stdio.h>
int G[50][50],x[50]; //G:adjacency matrix,x:colors
void next_color(int k){
int i,j;
x[k]=1; //coloring vertex with color1
for(i=0;i<k;i++){ //checking all
if(G[i][k]!=0 && x[k]==x[i])
x[k]=x[i]+1; //assign higher color than x[i]
}
}
int main(){
int n,e,i,j,k,l;
printf("Enter no. of vertices : ");
scanf("%d",&n); //total vertices
printf("Enter no. of edges : ");
scanf("%d",&e); //total edges
for(i=0;i<n;i++)
for(j=0;j<n;j++)
G[i][j]=0; //assign 0 to all index of adjacency matrix
printf("Enter indexes where value is 1");
for(i=0;i<e;i++)
{
scanf("%d %d",&k,&l);
G[k][l]=1;
G[l][k]=1;
}
for(i=0;i<n;i++)
next_color(i); //coloring each vertex
printf("Colors of vertices -->\n");
for(i=0;i<n;i++) //displaying color of each vertex
printf("Vertex[%d] : %d\n",i+1,x[i]);
return 0;
}
```

Output:



```
Enter no. of vertices : 6
Enter no. of edges : 5
Enter indexes where value is 1
0 1
2 3
2 4
1 4
6 5
Colors of vertices -->
Vertex[1] : 1
Vertex[2] : 2
Vertex[3] : 1
Vertex[4] : 2
Vertex[5] : 2
Vertex[6] : 1

Process returned 0 (0x0)   execution time : 30.686 s
Press any key to continue.
```

Conclusion: Thus it is observed that the Complexity **of Colouring using Backtracking Approach** Problem is **O(m^V).**