wednesday
3-6-2020

DAA Assignment-3
UNIT-3

N.V.S.k.kalyani
17U41A0554
3rd B.Tech CSE-C

2 marks

1. Define

Ans E-node:- A live node whose children are currently beign expanded is called E-node.

Live node:- A node which has been generated and all of its children have not yet been generated is called Live node.

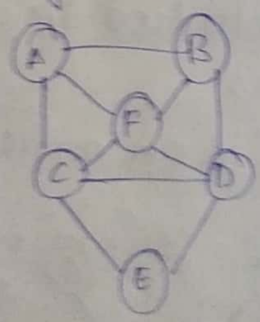
Dead node:- A dead node is a generated node which is not to be expanded further (or) all of its children has been generated.

2. Define

Ans Hamilton cycle:-

let G=(V,E) be a connected graph with 'n' vertices A hamilton (or) hamiltonian cycle is a path along n edge of G that visits every vertex

Ex:-



Hamiltonian cycle is :

A — B — D — E — C — F — A

3. write an algorithm for post-order traversal?

Ans Algorithm postorder (t)

```
{
  if(t ≠ 0) then
  {
    postorder (t → lchild);
```

174N1A0584

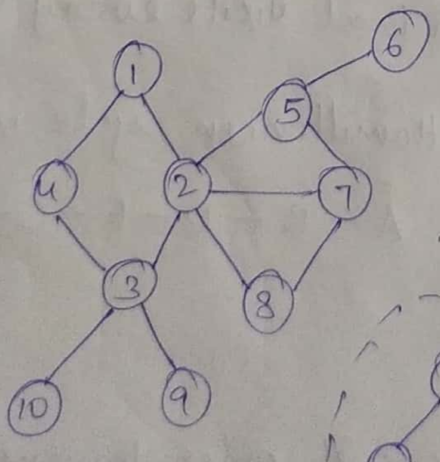
```
postorder(t -> rchild);
visit(t);
}
}
```

4. List out the applications of Backtracking?

Ans
* N-queen's problem
* Hamiltonian cycle problem
* Graph colouring problem
* subset-sum problem.

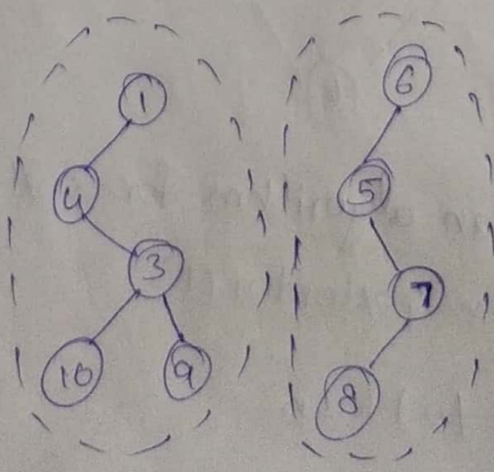
5. Define Articulation point?

Ans let G(V,E) be connected undirected graph, then an articulation point of graph G is a vertex, by deleting that vertext from the graph The graph is divided into two subgraphs.



∴ Deleting vertex 'x', it splits into two disjoint subgraph as

∴ '2' is an articulation point



=> Two disjoint graphs

10 marks

1a) Solve the 8 Queen's problem with an Example.

Ans 8 - Queen problem stated as :- consider a chess board of order 8x8. The problem is to place 8-Queen's on this board such that no two Queen can attack each other.

* The solution to 8 - Queen's can be obtained using backtracking method.

Ex:- consider the queen a[4,2]. The squares that are diagonal that runs from upper left to lower right are a[3,1], a[5,3], a[7,5], a[6,4], a[8,6].
All these squares have row - column value of 2

	1	2	3	4	5	6	7	8
1					(1,5)			
2				(2,4)				
3	(3,1)		(3,3)					
4		i,j (4,2)						
5	(5,1)		k,l (5,3)					
6				(6,4)				
7					(7,5)			
8						(8,6)		

row - column

3-1 = 2
4-2 = 2
5-2 = 2
6-4 = 2
7-5 = 2
8-6 = 2

row + column

5+1 = 6
4+2 = 6
3+3 = 6
2+4 = 6

5+1=6

* Also every element on the same diagonal (4,2) goes from the lower left to upper right has the same row + column value.

* Suppose two queens placed at position (i,j) and (k,l). They can same diagonal iff

$$\boxed{i-j=k-l} \quad \text{—} (1)$$

$$\boxed{i+j=k+l} \quad \text{—} (2)$$

From eq (1) ⇒ $j-l=i-k$

from eq (2) ⇒ $j-l=k-i$

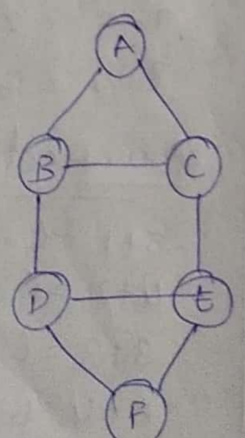
queens lie on the same diagonal if and only if

$$\boxed{|j-l|=|i-k|}$$

1b) construct an algorithm for graph coloring problem with a suitable example?

Ans Graph coloring:- let G be a graph and 'm' be a given positive Integer. Graph coloring is problem of coloring each vertex in a graph in a such a way that no two adjacent vertices have same color and only m-colors are used this problem is called m-coloring problem.

EX:-



There are 3-colors are used like Red, Green, Blue

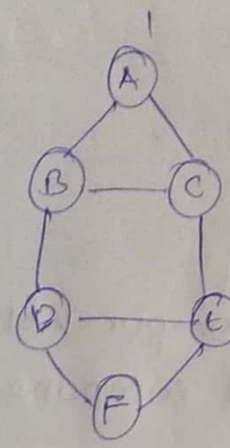
1 - Red color

2 - Green color

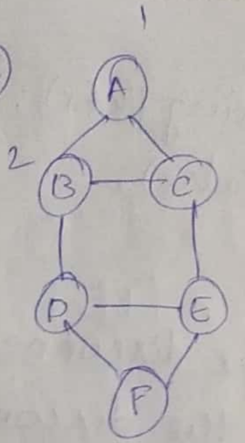
3 - Blue color

* Now color the graph, in such a way that no two adjacent vertices have same color and use only 3 colors.

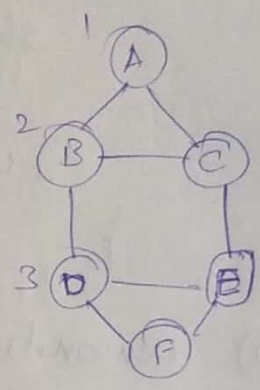
(1)



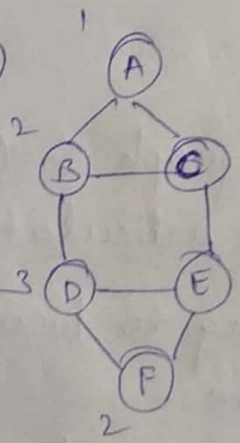
(2)



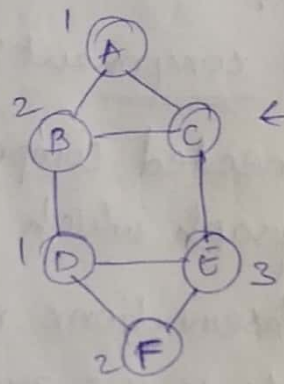
(3)



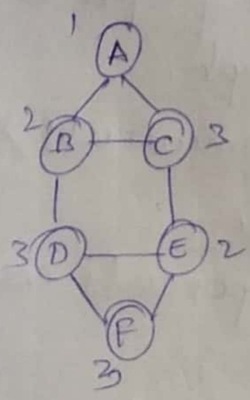
(4)



(5)



← struck there cannot assign 2 (or) 3 color Hence, we make backtrack



∴ Thus graph coloring problem is solved.

Algorithm :-

```
Algorithm mcoloring(k)
{
  repeat
  {
    next value(k);
    if (x[k]=0) then
    return;
    if(k=n) then
      write (x[1:n]);
```

```
    else
      mcoloring (k+1);
  } until (false);
}
```

② Demonstrate the disconnected components? How to determine the disconnected components using pseudocode?

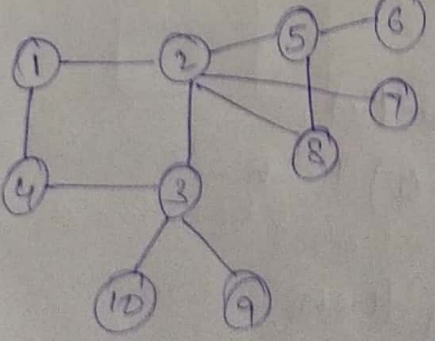
Ans Biconnected component:-

A biconnected component graph G=(V,E) is connected graph which has no articulation point.

* The key observations regarding to biconnected components of the graph are:-

1. A biconnected component of a graph G is maximal biconnected subgraph.
2. Two different biconnected components should not have any common edges.
3. The different biconnected components can have common vertex.
4. The common vertex which is attaching can have two (or) more biconnected components must have an articulation point of G.

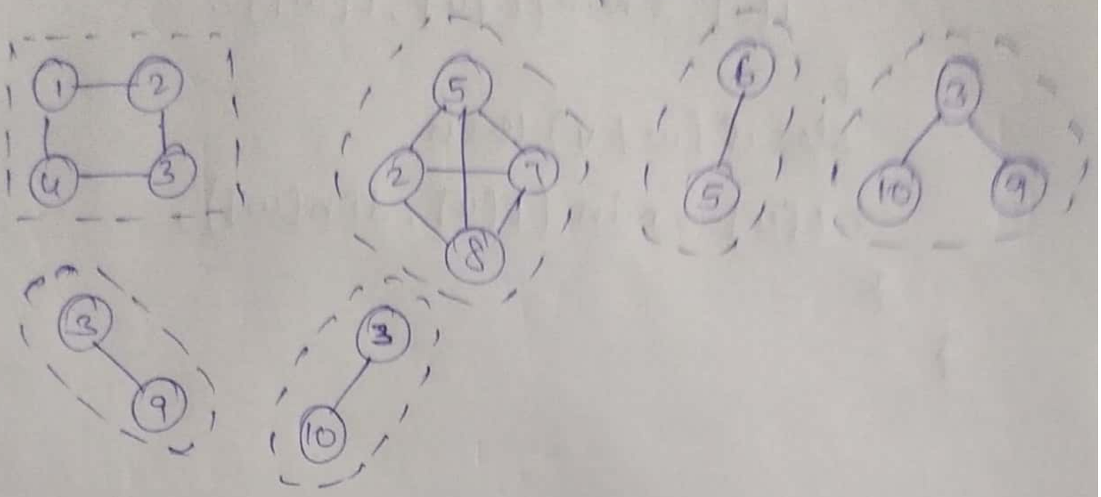
Ex:-



∴ Articulations points are 2, 3, 5

17UN1A0584

Hence the biconnected components are as follows



Algorithm Bicomp(u,v)

```
{
  dfn[u] := num;
  L[u] := num;
  num := num+1;
  for each vertex w adjacent from u do
  {
    if ((v ≠ w) and (dfn(w) < dfn[u])) then
      add (u,w) to top of a stack s;
    if (dfn[w]=0) then
    {
      if (L[w] ≥ dfn[u]) then
      {
        write("new bicomponent");
        repeat
        {
          Delete an edge from the top of stack s;
          let the edge be (x,y);
          write (x,y);
        }
        until(((x,y) = (u,w) or ((x,y) = (w,u)));
      }
      Bicomp(w,u);
```

174N1A0584

L[u] := min(L[u], L[w]);

}

else if (w ≠ v) then

L[u] := min(L[u], dfn[w]);

}

}

Scanned with CamScanner