

18.11 Depth first search using time stamps

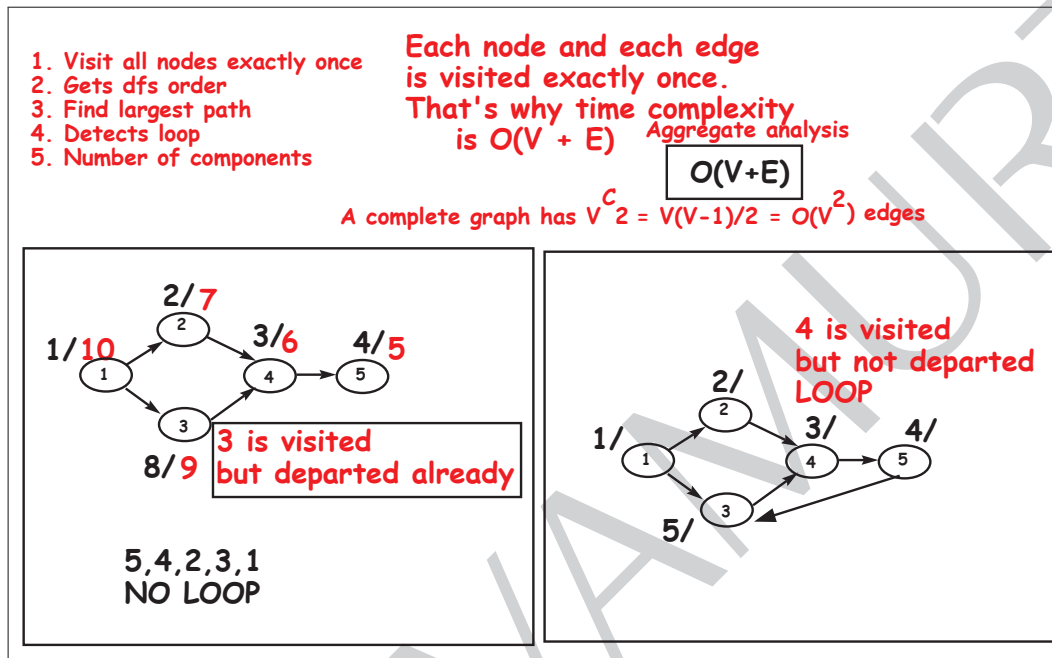


Figure 18.32: Depth first search on a directed graph using time stamps

18.11. DEPTH FIRST SEARCH USING TIME STAMPS

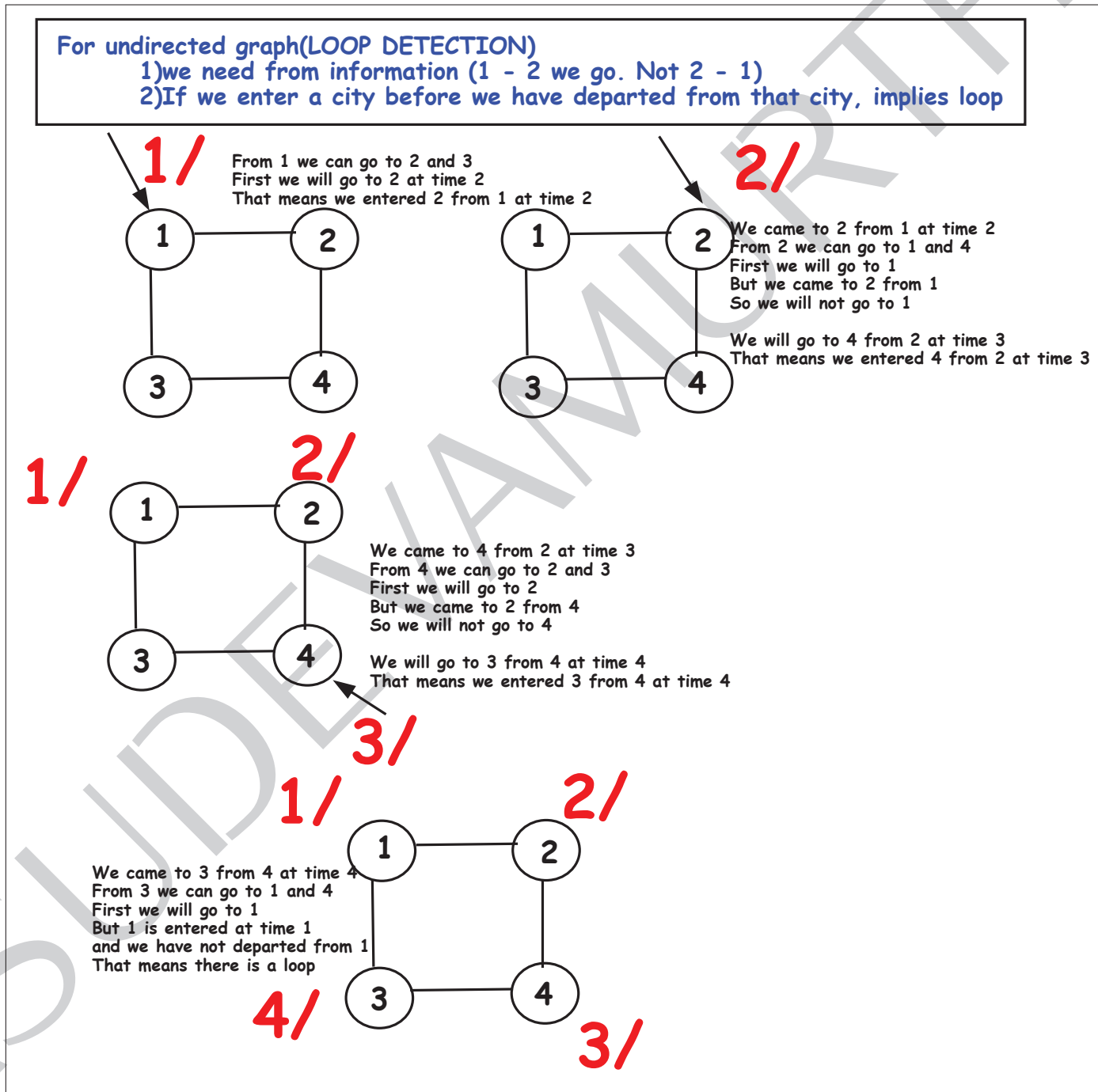


Figure 18.33: Depth first search on an undirected graph using time stamps and from

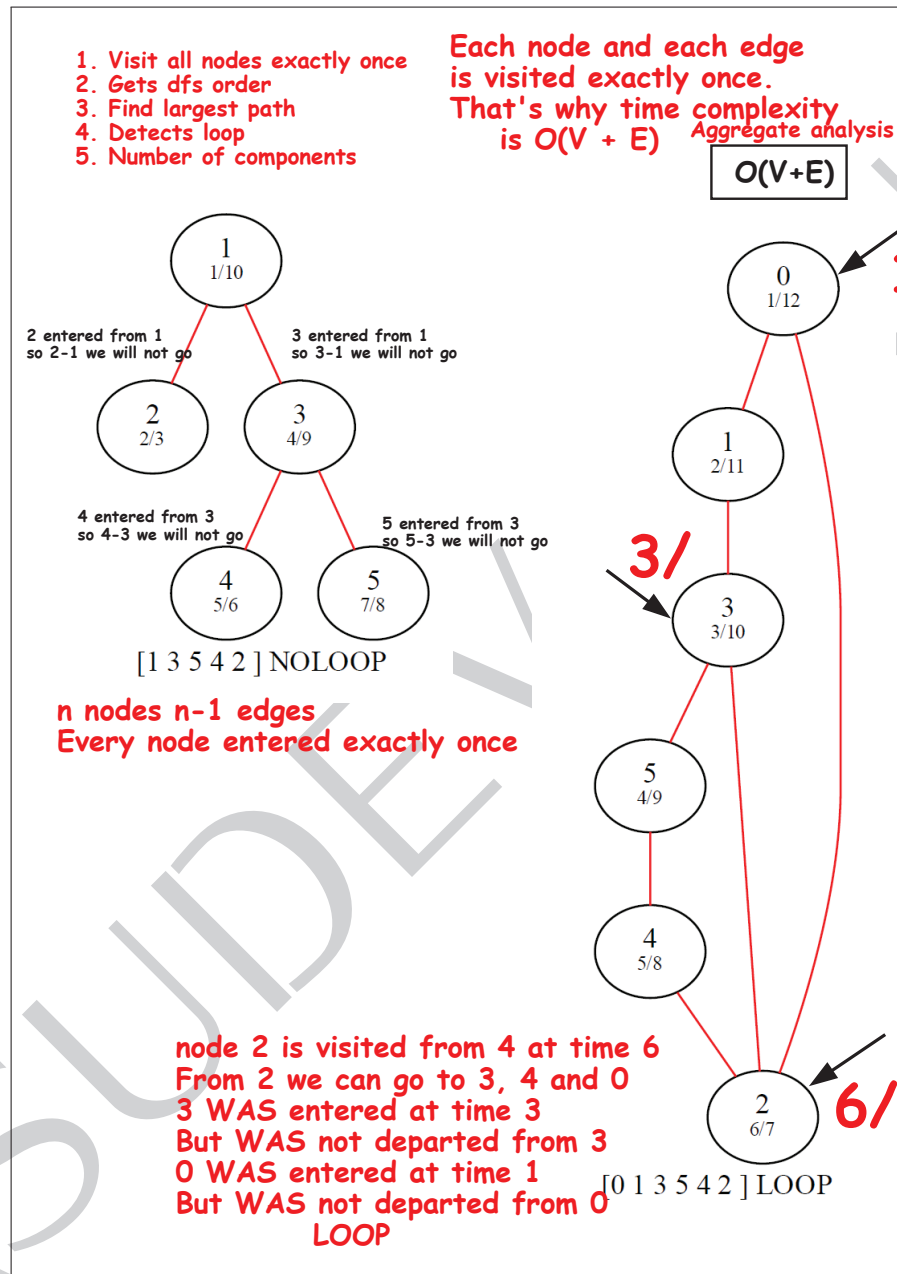


Figure 18.34: Depth first search on an undirected graph

18.11. DEPTH FIRST SEARCH USING TIME STAMPS

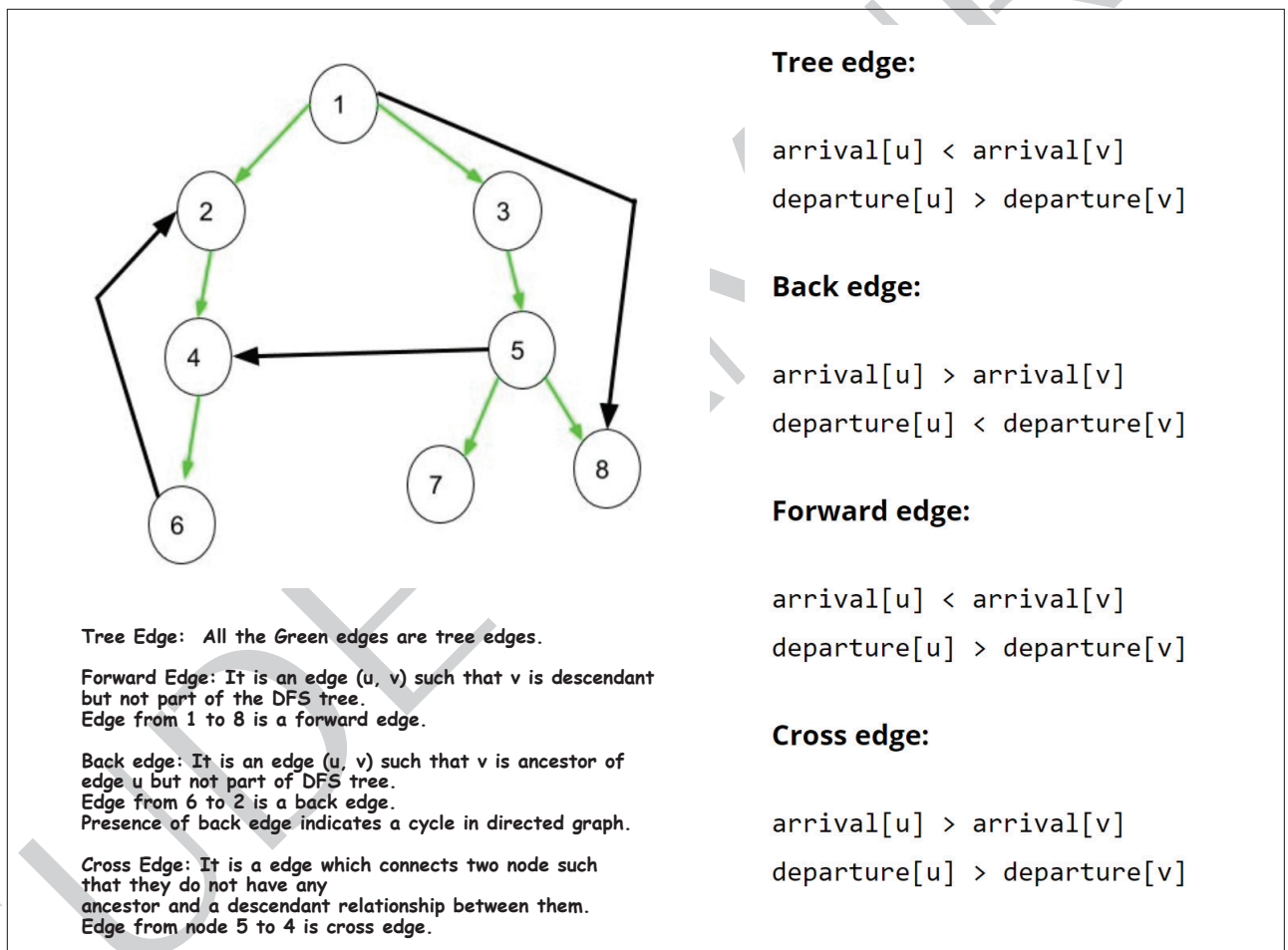


Figure 18.35: classification of edges in a directed graph

18.11.1 Depth first search on a undirected graph with no loop

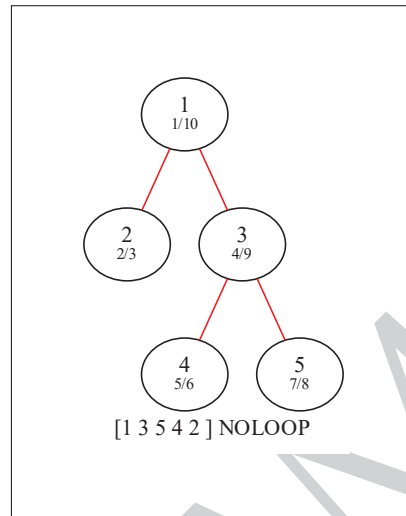


Figure 18.36: undirected graph with no loop

18.11. DEPTH FIRST SEARCH USING TIME STAMPS

```
## Jagadeesh Vasudevamurthy ####
## dot -Tpdf C:\scratch\outputs\dot\1dfs.dot -o C:\scratch\outputs\dot\1dfs.dot.pdf
digraph g {
    label = "[1 3 5 4 2 ] NOLOOP"
    1[label = <1<BR /><FONT POINT-SIZE="10">1/10</FONT>>]
    2[label = <2<BR /><FONT POINT-SIZE="10">2/3</FONT>>]
    3[label = <3<BR /><FONT POINT-SIZE="10">4/9</FONT>>]
    4[label = <4<BR /><FONT POINT-SIZE="10">5/6</FONT>>]
    5[label = <5<BR /><FONT POINT-SIZE="10">7/8</FONT>>]
    edge [dir=none, color=red]
    1 -> 2
    1 -> 3
    3 -> 4
    3 -> 5
}
```

18.11.2 Depth first search on a undirected graph with loop

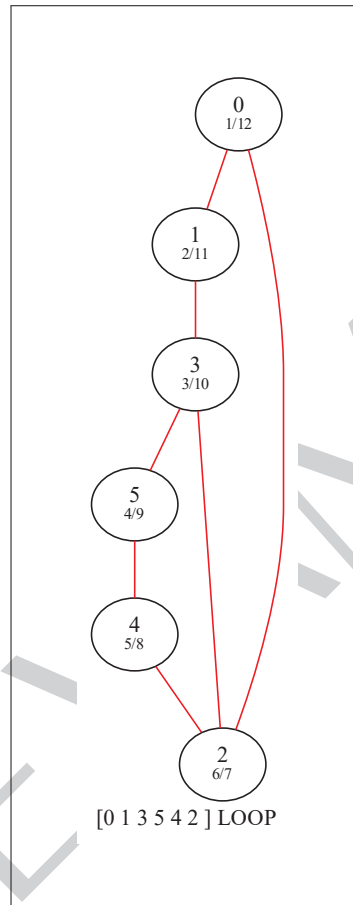


Figure 18.37: undirected graph with loop

18.11. DEPTH FIRST SEARCH USING TIME STAMPS

```
## Jagadeesh Vasudevamurthy ####
## dot -Tpdf C:\scratch\outputs\dot\udf1dfs.dot -o C:\scratch\outputs\dot\udf1dfs.dot.
digraph g {
    label = "[0 1 3 5 4 2 ] LOOP"
    0[label = <0<BR /><FONT POINT-SIZE="10">1/12</FONT>>]
    1[label = <1<BR /><FONT POINT-SIZE="10">2/11</FONT>>]
    3[label = <3<BR /><FONT POINT-SIZE="10">3/10</FONT>>]
    5[label = <5<BR /><FONT POINT-SIZE="10">4/9</FONT>>]
    4[label = <4<BR /><FONT POINT-SIZE="10">5/8</FONT>>]
    2[label = <2<BR /><FONT POINT-SIZE="10">6/7</FONT>>]
    edge [dir=none, color=red]
    0 -> 1
    0 -> 2
    1 -> 3
    3 -> 5
    3 -> 2
    5 -> 4
    4 -> 2
}
```


18.11.3 Depth first search on a directed graph with no loop

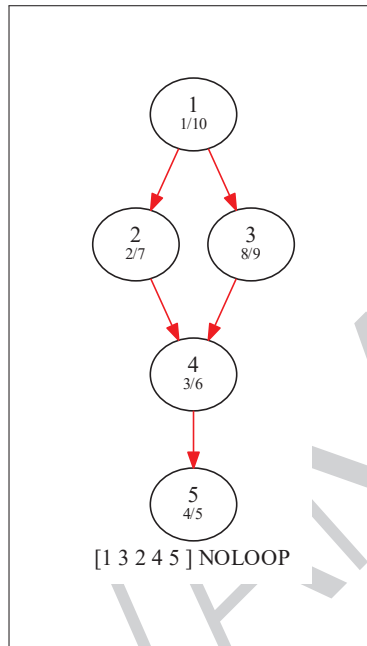


Figure 18.38: directed graph with no loop

18.11. DEPTH FIRST SEARCH USING TIME STAMPS

```
## Jagadeesh Vasudevamurthy ####
## dot -Tpdf C:\scratch\outputs\dot\2dfs.dot -o C:\scratch\outputs\dot\2dfs.dot.pdf
digraph g {
    label = "[1 3 2 4 5 ] NOLOOP"
    1[label = <1<BR /><FONT POINT-SIZE="10">1/10</FONT>>]
    2[label = <2<BR /><FONT POINT-SIZE="10">2/7</FONT>>]
    3[label = <3<BR /><FONT POINT-SIZE="10">8/9</FONT>>]
    4[label = <4<BR /><FONT POINT-SIZE="10">3/6</FONT>>]
    5[label = <5<BR /><FONT POINT-SIZE="10">4/5</FONT>>]
    edge [color=red]
    1 -> 2
    1 -> 3
    2 -> 4
    3 -> 4
    4 -> 5
}
```

18.11.4 Depth first search on a directed graph with loop

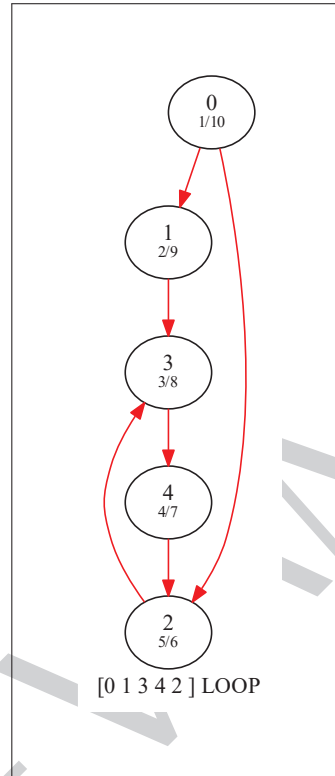


Figure 18.39: directed graph with loop

18.11. DEPTH FIRST SEARCH USING TIME STAMPS

```
## Jagadeesh Vasudevamurthy ####
## dot -Tpdf C:\scratch\outputs\dot\3dfs.dot -o C:\scratch\outputs\dot\3dfs.dot.pdf
digraph g {
    label = "[0 1 3 4 2 ] LOOP"
    0[label = <0<BR /><FONT POINT-SIZE="10">1/10</FONT>>]
    1[label = <1<BR /><FONT POINT-SIZE="10">2/9</FONT>>]
    2[label = <2<BR /><FONT POINT-SIZE="10">5/6</FONT>>]
    3[label = <3<BR /><FONT POINT-SIZE="10">3/8</FONT>>]
    4[label = <4<BR /><FONT POINT-SIZE="10">4/7</FONT>>]
    edge [color=red]
    0 -> 1
    0 -> 2
    1 -> 3
    2 -> 3
    3 -> 4
    4 -> 2
}
```

18.11.5 Depth first search on a directed graph with no loop

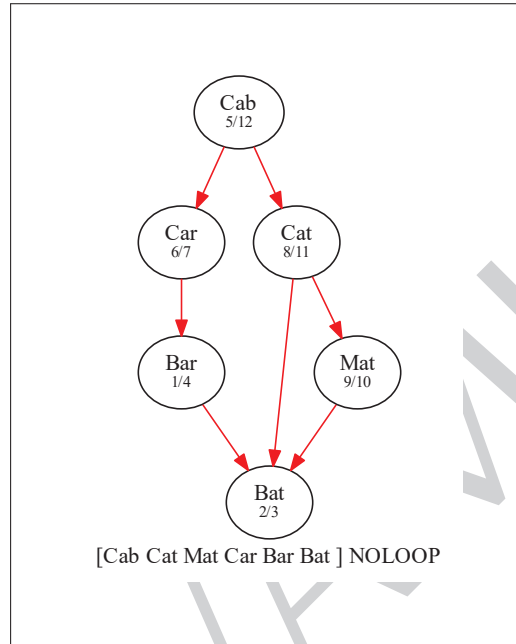


Figure 18.40: directed graph with no loop

18.11. DEPTH FIRST SEARCH USING TIME STAMPS

```
## Jagadeesh Vasudevamurthy ####
## dot -Tpdf C:\scratch\outputs\dot\catdfs.dot -o C:\scratch\outputs\dot\catdfs.dot.pdf
digraph g {
    label = "[Cab Cat Mat Car Bar Bat ] NOLOOP"
    Bar[label = <Bar<BR /><FONT POINT-SIZE="10">1/4</FONT>>]
    Bat[label = <Bat<BR /><FONT POINT-SIZE="10">2/3</FONT>>]
    Cab[label = <Cab<BR /><FONT POINT-SIZE="10">5/12</FONT>>]
    Car[label = <Car<BR /><FONT POINT-SIZE="10">6/7</FONT>>]
    Mat[label = <Mat<BR /><FONT POINT-SIZE="10">9/10</FONT>>]
    Cat[label = <Cat<BR /><FONT POINT-SIZE="10">8/11</FONT>>]
    edge [color=red]
    Bar -> Bat
    Cab -> Car
    Cab -> Cat
    Car -> Bar
    Mat -> Bat
    Cat -> Bat
    Cat -> Mat
}
```

18.11.6 Depth first search on a directed graph with no loop

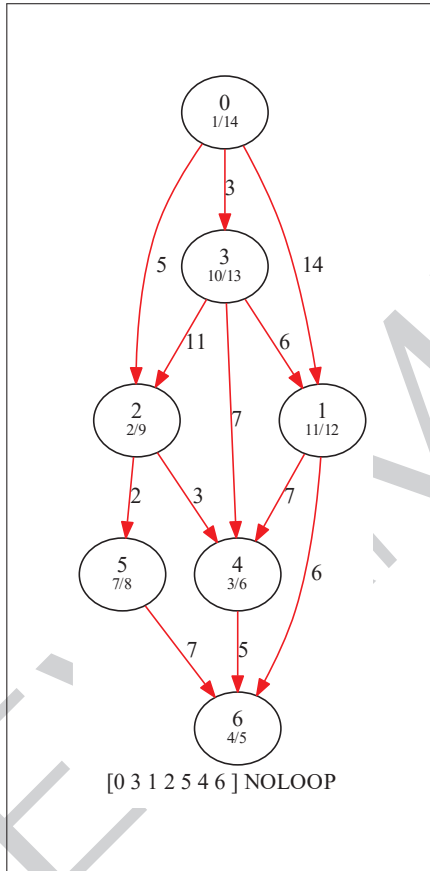


Figure 18.41: directed graph with no loop

18.11. DEPTH FIRST SEARCH USING TIME STAMPS

```
## Jagadeesh Vasudevamurthy ####
## dot -Tpdf C:\scratch\outputs\dot\7dfs.dot -o C:\scratch\outputs\dot\7dfs.dot.pdf
digraph g {
    label = "[0 3 1 2 5 4 6 ] NOLOOP"
    0[label = <0<BR /><FONT POINT-SIZE="10">1/14</FONT>>]
    2[label = <2<BR /><FONT POINT-SIZE="10">2/9</FONT>>]
    3[label = <3<BR /><FONT POINT-SIZE="10">10/13</FONT>>]
    1[label = <1<BR /><FONT POINT-SIZE="10">11/12</FONT>>]
    6[label = <6<BR /><FONT POINT-SIZE="10">4/5</FONT>>]
    4[label = <4<BR /><FONT POINT-SIZE="10">3/6</FONT>>]
    5[label = <5<BR /><FONT POINT-SIZE="10">7/8</FONT>>]
    edge [color=red]
    0 -> 2 [label = 5]
    0 -> 3 [label = 3]
    0 -> 1 [label = 14]
    2 -> 4 [label = 3]
    2 -> 5 [label = 2]
    3 -> 2 [label = 11]
    3 -> 1 [label = 6]
    3 -> 4 [label = 7]
    1 -> 6 [label = 6]
    1 -> 4 [label = 7]
    4 -> 6 [label = 5]
    5 -> 6 [label = 7]
}
```


18.11.7 Implementing depth first search

Implementing Depth First Search

```
class GraphDFSUsingTimeStamp{
    private Graph g ;
    private int [] work ;
    private boolean [] cycle;
    private ArrayList<Integer> topologicalOrderArray;
    private String f;

    GraphDFSUsingTimeStamp(Graph g, int [] work, boolean [] cycle, ArrayList<Integer> topologicalOrderArray, String f) {
        this.g = g ;
        this.work = work ;
        this.cycle = cycle ;
        this.topologicalOrderArray = topologicalOrderArray ;
        this.f = f ;
        //You MUST WRITE 2 routines
        dfs() ;
        writeDFSdot() ;
    }
}
```

GraphDFSUsingTimeStamp.java

```
public void dfsUsingTimeStamp(int [] work, boolean [] cycle, ArrayList<Integer> topologicalOrderArray, String f) {
    GraphDFSUsingTimeStamp g = new GraphDFSUsingTimeStamp(this, work, cycle, topologicalOrderArray, f);
}
```

Graph.java

```
private void dfsUsingTimeStamp(String name, GraphType.Type graphType, boolean expectedHasloop) {
    GraphIO io = new GraphIO();
    Graph g = new Graph(graphType, io);
    String f = name + ".txt" ;

    g.buildGraph(f) ;
    int [] work = {0};
    boolean [] cycle = {false} ;
    ArrayList<Integer> topologicalOrderArray = new ArrayList<Integer>();
    f = name + "dfs.dot" ;
    g.dfsUsingTimeStamp(work, cycle, topologicalOrderArray, f);
    boolean x = g.assertDFS(cycle[0], topologicalOrderArray);
}

private void testDfsUsingTimeStamp() {
    dfsUsingTimeStamp("u1", GraphType.Type.UNDIRECTED, false) ;
    dfsUsingTimeStamp("1", GraphType.Type.UNDIRECTED, false) ;
    dfsUsingTimeStamp("udf1", GraphType.Type.UNDIRECTED, true); //loop
    dfsUsingTimeStamp("2", GraphType.Type.DIRECTED, false);
    dfsUsingTimeStamp("3", GraphType.Type.DIRECTED, true); //loop
    dfsUsingTimeStamp("cat", GraphType.Type.DIRECTED, false);
    dfsUsingTimeStamp("7", GraphType.Type.WEIGHTED_DIRECTED, false);
    dfsUsingTimeStamp("mediumEWD", GraphType.Type.WEIGHTED_DIRECTED, true); //loop
}
```

GraphTest.java

Make sure topological order is correct using fanins

Figure 18.42: Implementing depth first search