

Breadth First Search (BFS) on graph (V,E) from root s

```
for (all u in V\{s})
{
    colour[u] = WHITE ;
    d[u]      = infinity;           // BFS-distance from s
    p[u]      = null;             // parent of u
}
colour[s] = GREY;  d[s] = 0 ; pa[s] = null;
Q = new Queue();  Q.enqueue(s); // put root on queue

while (Q not empty)
{
    u = Q.dequeue();
    for (all v in Adj(u)) // all neighbours v of u
    {                       // i.e. (u,v) in E
        if ( colour[v] == WHITE ) // v unvisited
        {
            colour[v] = GREY;
            d[v]      = d[u] + 1;
            p[v]      = u ;           // u BFS-parent of v
            Q.enqueue(v);           // put v at end of queue
        }
    }
    colour[u] = BLACK;
}
```

Depth First Search (DFS) on graph (V,E)

```
for (all u in V)
{
    colour[u] = WHITE ;
    p[u]      = null;          // parent of u
}

for (all u in V)
{
    if ( colour[u] == WHITE ) // u unvisited
        DFSvisit(u);
}
//-----

void DFSvisit( u )
{
    colour[u] = GREY;
    for (all v in Adj(u)) // all neighbours v of u
        if ( colour[v] == WHITE )
        {
            p[v] = u;
            DFSvisit(v); // recursive call
        }
    colour[u] = BLACK;
}
```

