Algorithmische Graphentheorie
SS 09

The backtracking approach of Brown

Procedure: Color(k)

```
{First possibility: assign v_k sequentially to one of the al-
ready existing color classes, if feasible.}
NrColors[k] := NrColors[k − 1];
{ determine the colors feasible for v_k:}
for i = 1 to NrColors[k] do
    FeasColor[k, i] := TRUE;
end for
for i: v_i ∈ Γ(v_k) do
    if i < k then
        FeasCol[k, c(i)] := FALSE;
    end if
end for
i = 1;
while (i ≤ NRColors[k]) ∧ (NrColors[k] < f) do
    if FeasColor[k, i] then
        c[k]:=i;
        if k < n then
            Color(k+1)
        else
            {found a better coloring!}
            f:=NrColors[k];
        end if
    end if
    i := i + 1;
end while
```

Procedure Color(k) - continued

```
{Second possibility: start up a new color class for v_k.}
if NrColors[k] + 1 < f then
    NrColors[k] := NrColors[k] + 1;
    c[k] := NrColors[k];
    if k < n then
        Color(k+1)
    else
        {found a better coloring!}
        f := NrColors[k];
    end if
end if
```

main program

```
f := n;      { Initialisation of upper bound on χ(G)}
order the nodes in a reasonable ordering v_1, v_2, ..., v_n;
c[1] := 1;
NrColors[1] := 1;
Color(2);
output f;
```