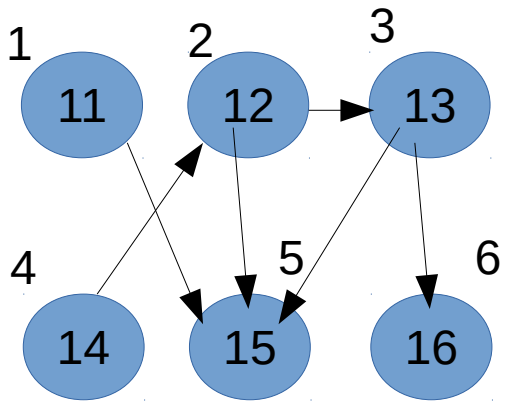
Grafo

```
int num_vertices;
list<vertices>* adj;
```



Output: 15,16,16,16,15,16

Vertices:

```
    int id;
    int nota_esperada;
    int nota_fim;
    //ou então só uma nota
    //cor para não repetir
```

Grafos:

```
	int num_vertices;
	list<vertices>* adj;
```

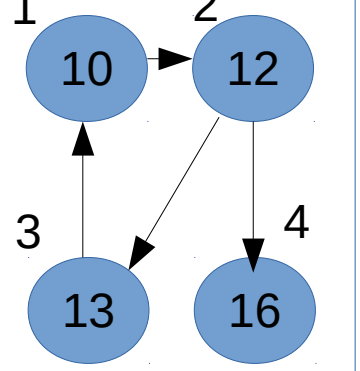
Vertices:

```
	int id;
	int nota_esperada;
	int nota_fim;
	//ou então só uma nota
	//cor para não repetir
(lista={white,gray,black})
```

Na primeira linha, faz-se new das variaveis antes a null (Vertices e Arestas)

IN

```
4,4
10
12
13
16
1,2
2,4
3,1
2,3
```



Out

```
16
16
13
16
```

Algoritmo Tarjan

TÉCNICO LISBOA

Algoritmo de Tarjan

low[u] = max[u]
Min = max

Cor = {white}

Cor = {white}

SCC_Tarjan(G)

```
visited ← 0
L ← 0
for u ∈ G.V do
    d[u] ← ∞
end for
for u ∈ G.V do
    if d[u] == ∞ then
        Tarjan_Visit(G, u)
    end if
end for
```

Tarjan_Visit(G,u)

```
d[u] ← low[u] ← visited
visited ← visited + 1
Push(L, u)
for v ∈ G.V do
    if d[v] == ∞ or v ∈ L then
        if d[v] == ∞ then
            Tarjan_Visit(G, v)
        end if
        low[u] ← min(low[u], low[v])
    end if
end for
if d[u] == low[u] then
    repeat
        v ← Pop(L)
    until u == v
end if
```

Cor = {white}

Algoritmo Tarjan

TÉCNICO LISBOA

Algoritmo de Tarjan

SCC_Tarjan(G)

```
visited ← 0
L ← 0
for u ∈ G.V do
    d[u] ← ∞
end for
for u ∈ G.V do
    if d[u] == ∞ then
        Tarjan_Visit(G, u)
    end if
end for
```

Tarjan_Visit(G,u)

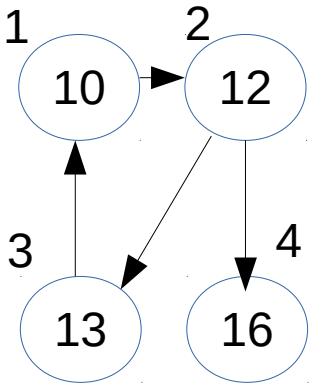
```
d[u] ← low[u] ← visited
visited ← visited + 1
Push(L, u)
for v ∈ G.V do
    if d[v] == ∞ or v ∈ L then
        if d[v] == ∞ then
            Tarjan_Visit(G, v)
        end if
        low[u] ← min(low[u], low[v])
    end if
end for
if d[u] == low[u] then
    repeat
        v ← Pop(L)
    until u == v
end if
```

Cor = {white}

low[u] = max[u]
Min = max

Cor = {white}

Cor = {white}



L= 1,

Cor = {white}

Cor = {white}

Algoritmo Tarjan

TÉCNICO LISBOA

Algoritmo de Tarjan

SCC_Tarjan(G)

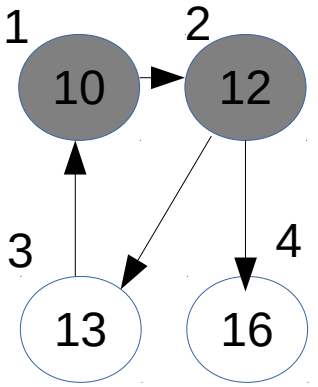
```
visited ← 0
L ← 0
for u ∈ G.V do
    d[u] ← ∞
end for
for u ∈ G.V do
    if d[u] == ∞ then
        Tarjan_Visit(G, u)
    end if
end for
```

Tarjan_Visit(G,u)

```
d[u] ← low[u] ← visited
visited ← visited + 1
Push(L, u)
for v ∈ G.V do
    if d[v] == ∞ or v ∈ L then
        if d[v] == ∞ then
            Tarjan_Visit(G, v)
        end if
        low[u] ← min(low[u], low[v])
    end if
end for
if d[u] == low[u] then
    repeat
        v ← Pop(L)
    until u == v
end if
```

Cor = {white}

low[u] = max[u]
Min = max



L= 1, 2,

Cor = {white}

Cor = {white}

Algoritmo Tarjan

TÉCNICO LISBOA

Algoritmo de Tarjan

SCC_Tarjan(G)

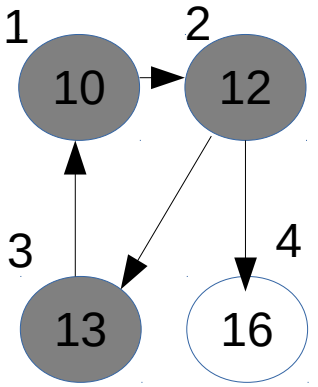
```
visited ← 0
L ← 0
for u ∈ G.V do
    d[u] ← ∞
end for
for u ∈ G.V do
    if d[u] == ∞ then
        Tarjan_Visit(G, u)
    end if
end for
```

Tarjan_Visit(G,u)

```
d[u] ← low[u] ← visited
visited ← visited + 1
Push(L, u)
for v ∈ G.V do
    if d[v] == ∞ or v ∈ L then
        if d[v] == ∞ then
            Tarjan_Visit(G, v)
        end if
        low[u] ← min(low[u], low[v])
    end if
end for
if d[u] == low[u] then
    repeat
        v ← Pop(L)
    until u == v
end if
```

low[u] = max[u]
Min = max

Cor = {white}



L= 1, 2, 3

Algoritmo Tarjan

TÉCNICO LISBOA

Algoritmo de Tarjan

SCC_Tarjan(G)

```
visited ← 0
L ← 0
for u ∈ G.V do
    d[u] ← ∞
end for
for u ∈ G.V do
    if d[u] == ∞ then
        Tarjan_Visit(G, u)
    end if
end for
```

Tarjan_Visit(G,u)

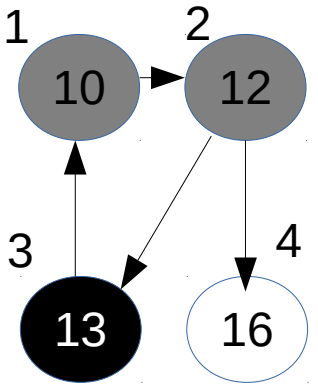
```
d[u] ← low[u] ← visited
visited ← visited + 1
Push(L, u)
for v ∈ G.V do
    if d[v] == ∞ or v ∈ L then
        if d[v] == ∞ then
            Tarjan_Visit(G, v)
        end if
        low[u] ← min(low[u], low[v])
    end if
end for
if d[u] == low[u] then
    repeat
        v ← Pop(L)
    until u == v
end if
```

Cor = {white}

low[u] = max[u]
Min = max

Cor = {white}

Cor = {white}



L= 1, 2, ~~3,~~

Algoritmo Tarjan

TÉCNICO LISBOA

Algoritmo de Tarjan

SCC_Tarjan(G)

```
visited ← 0
L ← 0
for u ∈ G.V do
    d[u] ← ∞
end for
for u ∈ G.V do
    if d[u] == ∞ then
        Tarjan_Visit(G, u)
    end if
end for
```

Tarjan_Visit(G,u)

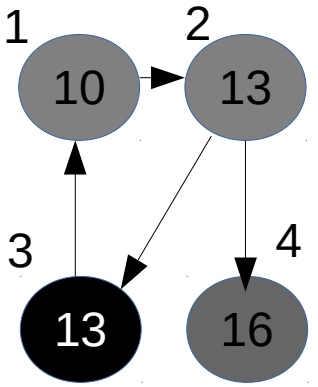
```
d[u] ← low[u] ← visited
visited ← visited + 1
Push(L, u)
for v ∈ G.V do
    if d[v] == ∞ or v ∈ L then
        if d[v] == ∞ then
            Tarjan_Visit(G, v)
        end if
        low[u] ← min(low[u], low[v])
    end if
end for
if d[u] == low[u] then
    repeat
        v ← Pop(L)
    until u == v
end if
```

Cor = {white}

low[u] = max[u]
Min = max

Cor = {white}

Cor = {white}



L= 1, 2, ~~3~~, 4

Algoritmo Tarjan

TÉCNICO LISBOA

Algoritmo de Tarjan

SCC_Tarjan(G)

```
visited ← 0
L ← 0
for u ∈ G.V do
    d[u] ← ∞
end for
for u ∈ G.V do
    if d[u] == ∞ then
        Tarjan_Visit(G, u)
    end if
end for
```

Tarjan_Visit(G,u)

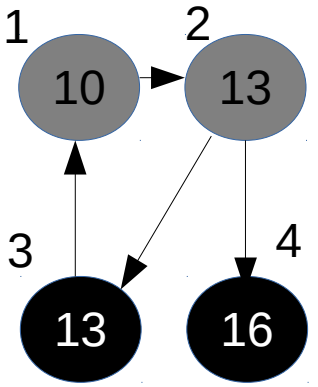
```
d[u] ← low[u] ← visited
visited ← visited + 1
Push(L, u)
for v ∈ G.V do
    if d[v] == ∞ or v ∈ L then
        if d[v] == ∞ then
            Tarjan_Visit(G, v)
        end if
        low[u] ← min(low[u], low[v])
    end if
end for
if d[u] == low[u] then
    repeat
        v ← Pop(L)
    until u == v
end if
```

low[u] = max[u]
Min = max

Cor = {white}

Cor = {white}

Cor = {white}



L= 1, 2, ~~3, 4~~

Algoritmo Tarjan

TÉCNICO LISBOA

Algoritmo de Tarjan

SCC_Tarjan(G)

```
visited ← 0
L ← 0
for u ∈ G.V do
    d[u] ← ∞
end for
for u ∈ G.V do
    if d[u] == ∞ then
        Tarjan_Visit(G, u)
    end if
end for
```

Tarjan_Visit(G,u)

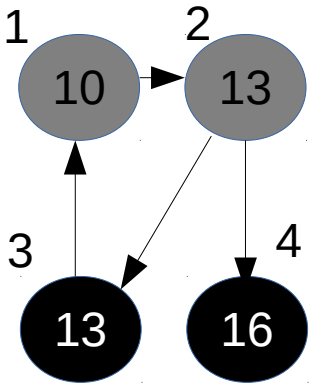
```
d[u] ← low[u] ← visited
visited ← visited + 1
Push(L, u)
for v ∈ G.V do
    if d[v] == ∞ or v ∈ L then
        if d[v] == ∞ then
            Tarjan_Visit(G, v)
        end if
        low[u] ← min(low[u], low[v])
    end if
end for
if d[u] == low[u] then
    repeat
        v ← Pop(L)
    until u == v
end if
```

low[u] = max[u]
Min = max

Cor = {white}

Cor = {white}

Cor = {white}



L= 1, ~~2, 3, 4~~

Algoritmo Tarjan

TÉCNICO LISBOA

Algoritmo de Tarjan

SCC_Tarjan(G)

```
visited ← 0
L ← 0
for u ∈ G.V do
    d[u] ← ∞
end for
for u ∈ G.V do
    if d[u] == ∞ then
        Tarjan_Visit(G, u)
    end if
end for
```

Tarjan_Visit(G,u)

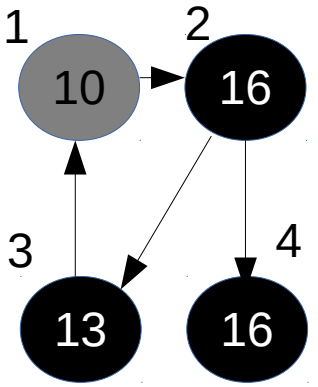
```
d[u] ← low[u] ← visited
visited ← visited + 1
Push(L, u)
for v ∈ G.V do
    if d[v] == ∞ or v ∈ L then
        if d[v] == ∞ then
            Tarjan_Visit(G, v)
        end if
        low[u] ← min(low[u], low[v])
    end if
end for
if d[u] == low[u] then
    repeat
        v ← Pop(L)
    until u == v
end if
```

Cor = {white}

low[u] = max[u]
Min = max

Cor = {white}

Cor = {white}



L= 1, ~~2, 3, 4~~

Algoritmo Tarjan

TÉCNICO LISBOA

Algoritmo de Tarjan

SCC_Tarjan(G)

```
visited ← 0
L ← 0
for u ∈ G.V do
    d[u] ← ∞
end for
for u ∈ G.V do
    if d[u] == ∞ then
        Tarjan_Visit(G, u)
    end if
end for
```

Tarjan_Visit(G,u)

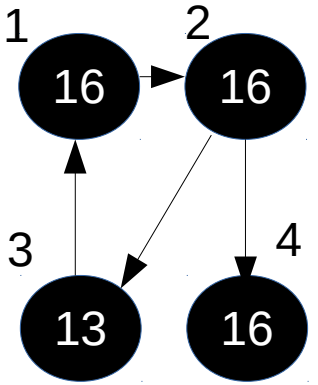
```
d[u] ← low[u] ← visited
visited ← visited + 1
Push(L, u)
for v ∈ G.V do
    if d[v] == ∞ or v ∈ L then
        if d[v] == ∞ then
            Tarjan_Visit(G, v)
        end if
        low[u] ← min(low[u], low[v])
    end if
end for
if d[u] == low[u] then
    repeat
        v ← Pop(L)
    until u == v
end if
```

Cor = {white}

low[u] = max[u]
Min = max

Cor = {white}

Cor = {white}



L= ~~1, 2, 3, 4~~

Algoritmo Tarjan

TÉCNICO LISBOA

Algoritmo de Tarjan

SCC_Tarjan(G)

```
visited ← 0
L ← 0
for u ∈ G.V do
    d[u] ← ∞
end for
for u ∈ G.V do
    if d[u] == ∞ then
        Tarjan_Visit(G, u)
    end if
end for
```

Tarjan_Visit(G,u)

```
d[u] ← low[u] ← visited
visited ← visited + 1
Push(L, u)
for v ∈ G.V do
    if d[v] == ∞ or v ∈ L then
        if d[v] == ∞ then
            Tarjan_Visit(G, v)
        end if
        low[u] ← min(low[u], low[v])
    end if
end for
if d[u] == low[u] then
    repeat
        v ← Pop(L)
    until u == v
end if
```

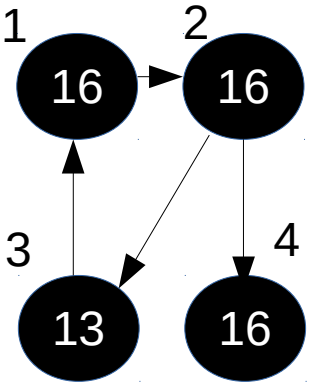
P.T. Monteiro ASA @ LEIC-T 2019/2020 14/42

low[u] = max[u]
Min = max

Cor = {white}

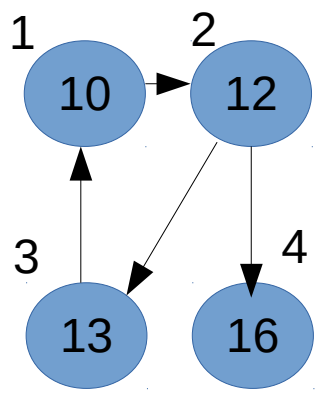
Cor = {white}

Cor = {white}



IN

4,4
10
12
13
16
1,2
2,4
3,1
2,3



16
16
13
16

L= ~~1, 2, 3, 4~~