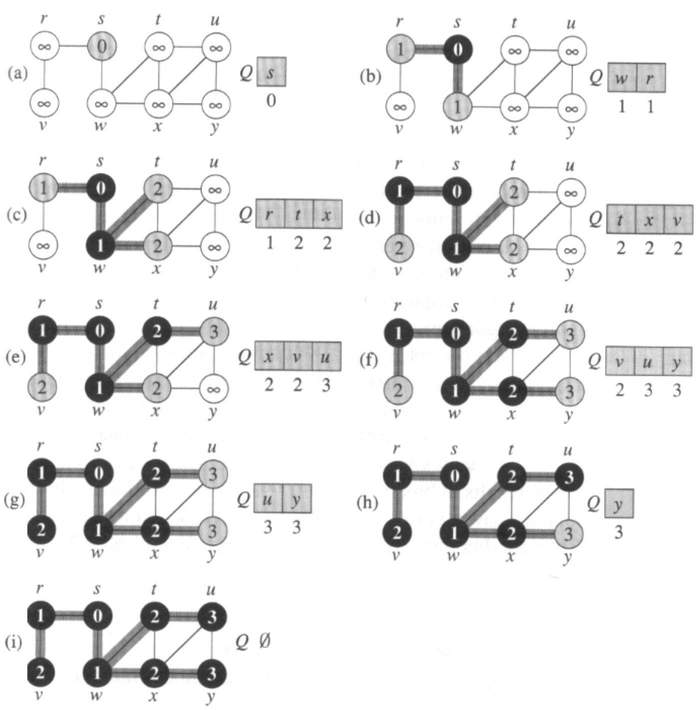
Búsqueda en Anchura (BFS) [Cormen]

```
BFS( G, s ) . . . . . . . . . . . . . . . . . . . . . . . . . . . O(V+E)
for each vertex u ∈ V[ G ] – {s}
   do color[ u ] ← WHITE
      d[ u ] ← ∞
      π[ u ] ← NIL
color[ s ] ← GRAY
d[ s ] ← 0
π[ s ] ← NIL
Q ← Ø
ENQUEUE( Q, s )
while Q ≠ Ø
   do u ← DEQUEUE( Q )
      for each v ∈ Adj[ u ]
         do if color[ v ] = WHITE
            then color[ v ] ← GRAY
                 d[ v ] ← d[ u ] + 1
                 π[ v ] ← u
                 ENQUEUE( Q, v )
      color[ u ] ← BLACK
```

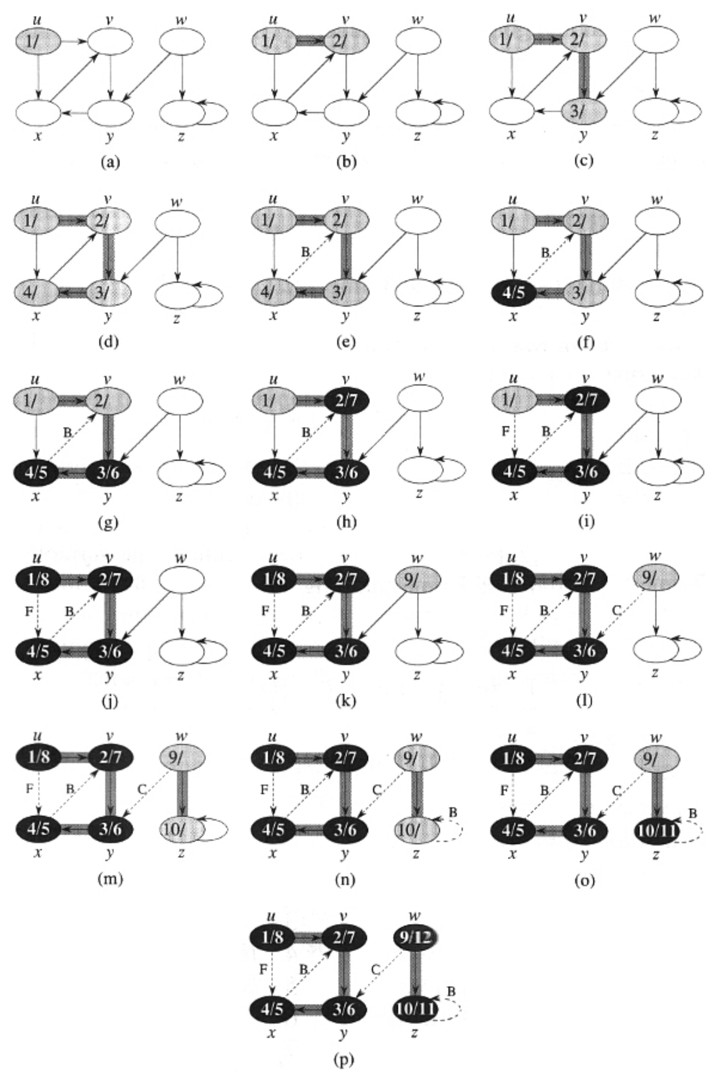


```
PRINT-PATH( G, s, v )
if v = s
   then print s
   else if π[ v ] = NIL
      then print "no path from" s "to" v "exists"
      else PRINT-PATH( G, s, π[ v ] )
           print v
```

Búsqueda en Profundidad (DFS) [Cormen]

```
DFS( G ) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Θ(V+E)
for each vertex u ∈ V[ G ]
   do color[ u ] ← WHITE
      π[ u ] ← NIL
time ← 0
for each vertex u ∈ V[ G ]
   do if color[ u ] = WHITE
      then DFS-VISIT( u )
```

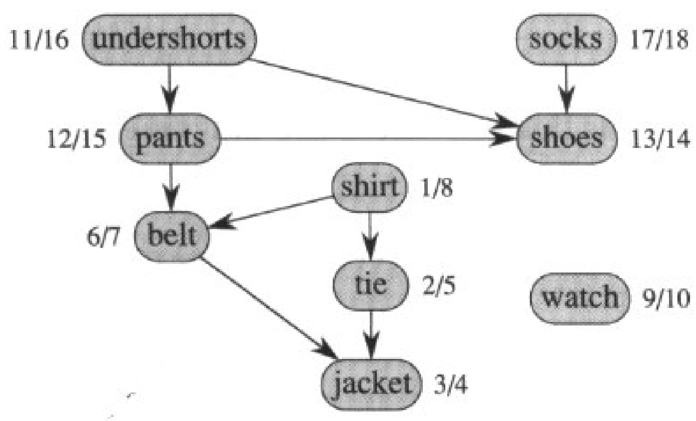
```
DFS-VISIT( u )
color[ u ] ← GRAY
time ← time + 1
d[ u ] ← time
for each v ∈ Adj[ u ]
   do if color[ v ] = WHITE
      then π[ v ] ← u
           DFS-VISIT( v )
color[ u ] ← BLACK
f [ u ] ← time ← time + 1
```

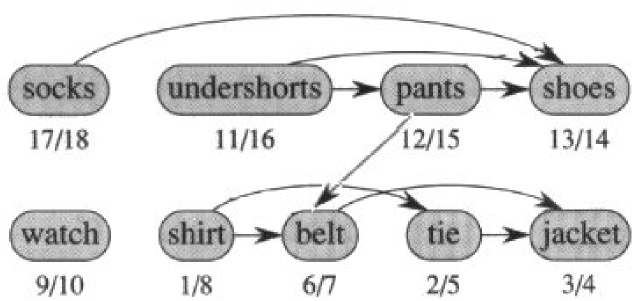


Topological Sort [Cormen]

Topological-Sort(G) . . . . Θ(V+E)

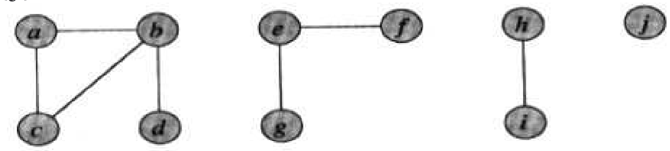
1. call DFS(G) to compute finishing times $f[v]$ for each vertex v
2. as each vertex is finished, insert it onto the front of a linked list
3. **return** the linked list of vertices





Connected Components(G)

A continuación se muestra un grafo con cuatro componentes conexas: {a, b, c, d}, {e, f, g}, {h, i} y {j}.



Edge processed	Collection of disjoint sets									
initial sets	{a}	{b}	{c}	{d}	{e}	{f}	{g}	{h}	{i}	{j}
(b,d)	{a}	{b,d}	{c}		{e}	{f}	{g}	{h}	{i}	{j}
(e,g)	{a}	{b,d}	{c}		{e,g}	{f}		{h}	{i}	{j}
(a,c)	{a,c}	{b,d}			{e,g}	{f}		{h}	{i}	{j}
(h,i)	{a,c}	{b,d}			{e,g}	{f}		{h,i}		{j}
(a,b)	{a,b,c,d}				{e,g}	{f}		{h,i}		{j}
(e,f)	{a,b,c,d}				{e,f,g}			{h,i}		{j}
(b,c)	{a,b,c,d}				{e,f,g}			{h,i}		{j}

Connected-Components(G)

1. **for** each vertex $v \in V[G]$
2. **do** Make-Set(v)
3. **for** each edge $(u, v) \in E[G]$
4. **do if** Find-Set(u) $\neq$ Find-Set(v)
5. **then** Union(u, v)

Same-Component(u, v)

1. **if** Find-Set(u) = Find-Set(v)
2. **then return** TRUE
3. **else return** FALSE

Estructura de datos: Union-Find

Make-set(x)

1. $p[x] \leftarrow x$
2. $rank[x] \leftarrow 0$

Union(x, y)

1. Link(Find-Set(x), Find-Set(y))

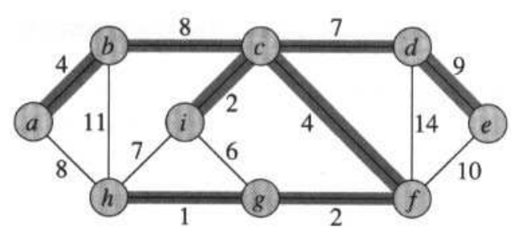
Link(x, y)

1. **if** $rank[x] > rank[y]$
2. **then** $p[y] \leftarrow x$
3. **else** $p[x] \leftarrow y$
4. **if** $rank[x] = rank[y]$
5. **then** $rank[y] \leftarrow rank[y] + 1$

Find-Set(x)

1. **if** $x \neq p[x]$
2. **then** $p[x] \leftarrow$ Find-Set($p[x]$)
3. **return** $p[x]$

Minimum Spanning Tree (MST)



MST-Kruskal(G, w)

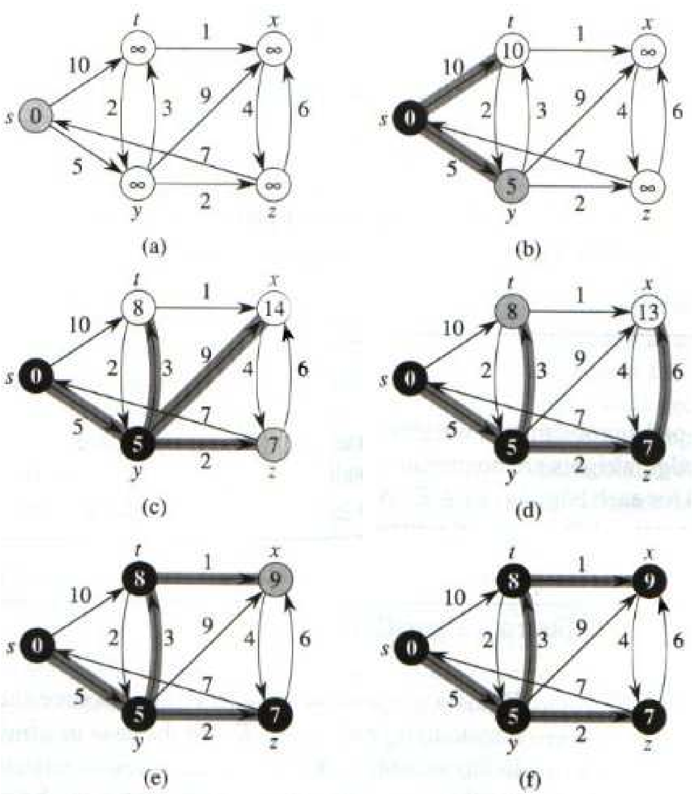
1. A $\leftarrow \emptyset$
2. **for** each vertex $v \in V[G]$
3. **do** Make-Set(v)
4. sort the edges of E into nondecreasing order by weight w
5. **for** each edge $(u, v) \in E$, taken in nondecreasing order by weight
6. **do if** Find-Set(u) $\neq$ Find-Set(v)
7. **then** A $\leftarrow$ A $\cup$ { (u, v) }
8. Union (u, v)
9. **return** A

DIJKSTRA

```
INITIALIZE-SINGLE-SOURCE( G, s )
for each vertex v ∈ V[ G ]
    do d[ v ] ← ∞
       π[ v ] ← NIL
d[ s ] ← 0
```

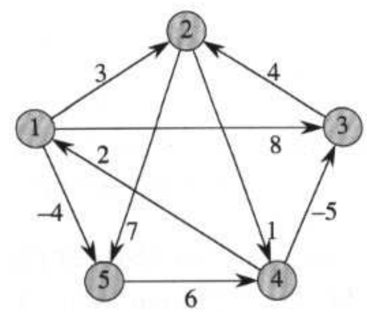
```
RELAX( u, v, w )
if d[ v ] > d[ u ] + w( u, v )
    then d[ v ] ← d[ u ] + w( u, v )
         π[ v ] ← u
```

```
Dijkstra( G, w, s )
INITIALIZE-SINGLE-SOURCE( G, s )
S ← Ø
Q ← V[ G ]
while Q ≠ Ø
    do u ← EXTRACT-MIN( Q )
       S ← S ∪ { u }
       for each vertex v ∈ Adj[ u ]
           do RELAX( u, v, w )
```



FLOYD-WARSHALL

```
FLOYD-WARSHALL( W )
n ← rows( W )
for i ←1 to n
    do for j ← 1 to n
        do if i = j or W[ i ][ j ] = 0
            D[ i ][ j ] ← ∞
            π[ i ][ j ] ← NIL
        else if W[ i ][ j ] > 0
            D[ i ][ j ] ← W[ i ][ j ]
            π[ i ][ j ] ← i
for k ← 1 to n
    do for i ← 1 to n
        do for j ← 1 to n
            do if D[ i ][ j ] > D[ i ][ k ] + D[ k ][ j ]
                D[ i ][ j ] ← D[ i ][ k ] + D[ k ][ j ]
                π[ i ][ j ] ← k
```



$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(0)} = \begin{pmatrix} \text{NIL} & 1 & 1 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & \text{NIL} \\ 4 & \text{NIL} & 4 & \text{NIL} & \text{NIL} \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(1)} = \begin{pmatrix} \text{NIL} & 1 & 1 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & \text{NIL} \\ 4 & 1 & 4 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(2)} = \begin{pmatrix} \text{NIL} & 1 & 1 & 2 & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & 2 & 2 \\ 4 & 1 & 4 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(3)} = \begin{pmatrix} \text{NIL} & 1 & 1 & 2 & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & 2 & 2 \\ 4 & 3 & 4 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \quad \Pi^{(4)} = \begin{pmatrix} \text{NIL} & 1 & 4 & 2 & 1 \\ 4 & \text{NIL} & 4 & 2 & 1 \\ 4 & 3 & \text{NIL} & 2 & 1 \\ 4 & 3 & 4 & \text{NIL} & 1 \\ 4 & 3 & 4 & 5 & \text{NIL} \end{pmatrix}$$

$$D^{(5)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \quad \Pi^{(5)} = \begin{pmatrix} \text{NIL} & 3 & 4 & 5 & 1 \\ 4 & \text{NIL} & 4 & 2 & 1 \\ 4 & 3 & \text{NIL} & 2 & 1 \\ 4 & 3 & 4 & \text{NIL} & 1 \\ 4 & 3 & 4 & 5 & \text{NIL} \end{pmatrix}$$