```
DFS(G)
for ogni vertice u ∈ V[G]
    do color[u] ← WHITE
       π[u] ← NIL
time ← 0
for ogni vertice u ∈ V[G]
    do if color[u] = WHITE
         then DFS-VISIT(u)
```

```
DFS-VISIT(u)
color[u] ← GRAY                    ▷ Il vertice bianco u è stato appena scoperto
d[u] ← time ← time + 1
for ogni v ∈ Adj[u]                ▷ Si esplora l'arco (u, v)
    do if color[v] = WHITE
          then π[v] ← u
               DFS-VISIT(v)
color[u] ← BLACK                   ▷ Si rende u nero: la sua visita è finita.
f[u] ← time ← time + 1
```

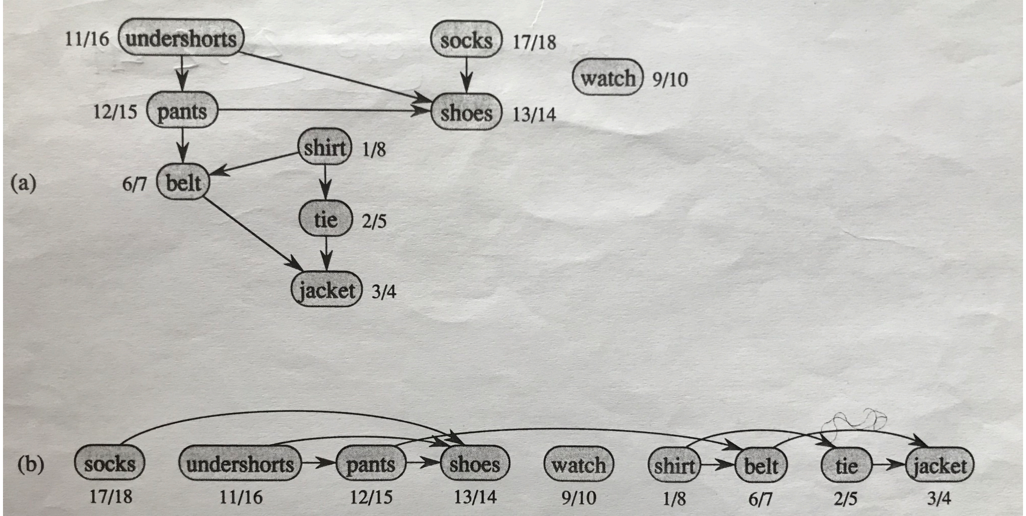
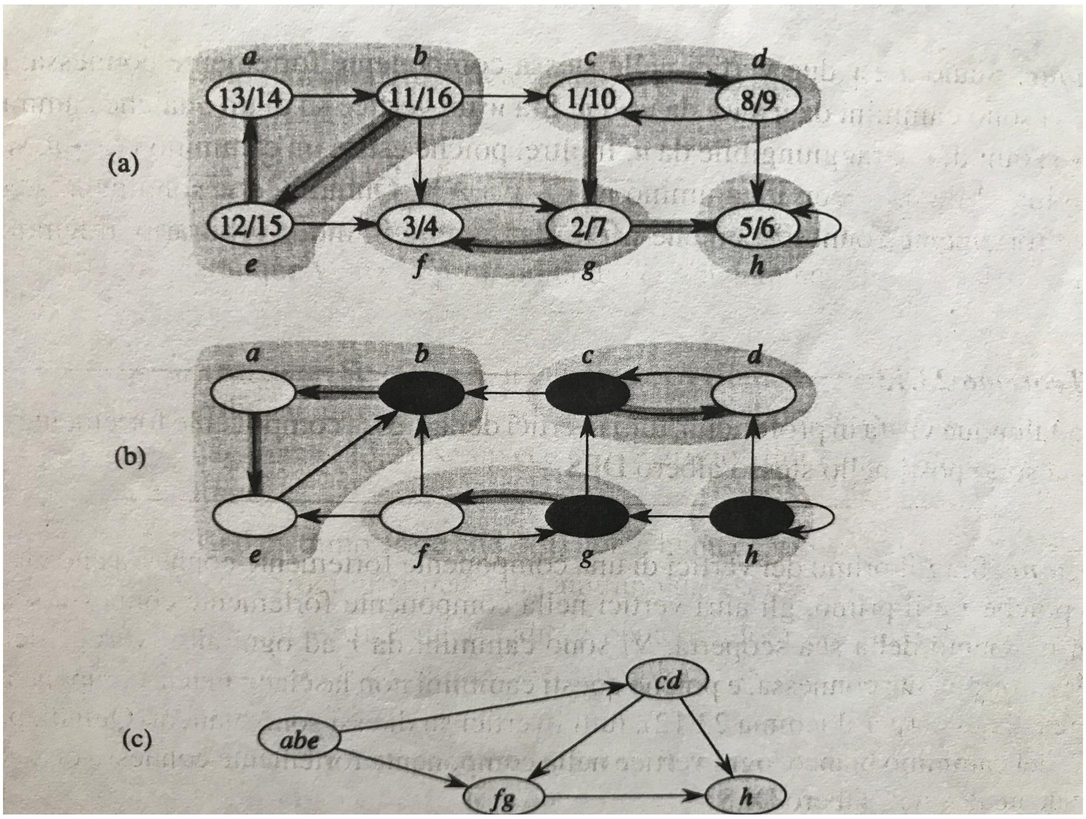


Figure 23.7 **(a)** Professor Bumstead topologically sorts his clothing when getting dressed. Each directed edge (u, v) means that garment u must be put on before garment v. The discovery and finishing times from a depth-first search are shown next to each vertex. **(b)** The same graph shown topologically sorted. Its vertices are arranged from left to right in order of decreasing finishing time. Note that all directed edges go from left to right.



Strongly-Connected-Components(G)

1 chiama DFS(G) per calcolare i tempi di fine visita $f[u]$ per ogni vertice u

2 calcola G^T

3 chiama DFS(G^T), ma nel ciclo principale di DFS, considera i vertici in ordine decrescente di $f[u]$ (come calcolati nella linea 1)

4 dai in output i vertici di ogni albero nella foresta DFS prodotta dal passo 3 come una diversa componente fortemente connessa

Consistenza e soluzione di un CSP in PA

1. Trasforma i vincoli: $x = y$ in $x \leq y$ e $y \leq x$, $x > y$ in $y < x$, $x \geq y$ in $y \leq x$
2. Crea grafo **G** *diretto* dei vincoli omettendo $x \neq y$ (solo relazioni $<$, $\leq$)
3. Se **G è un DAG** il CSP è consistente: usa DFS per trovare *ordinamento* delle variabili (attraverso ordinamento topologico di **G**) e deriva *soluzione*
4. Se **G ha cicli** (non è un DAG):
   a. Calcola gli SCC di **G**
   b. Se esiste un SCC con due nodi x e y collegati da < o nel CSP c'è $x \neq y$: *CSP Inconsistente*
   c. Altrimenti **collassa** ogni SCC in un singolo nodo generando un **DAG G'.** [FARE ESEMPIO di collassamento]
   d. Trova ordinamento variabili attarverso ordinamento topologico di **G'** (DFS) e soluzione

NB: se non so che G è un DAG passo da 2 direttamente a 4.a