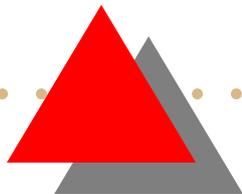


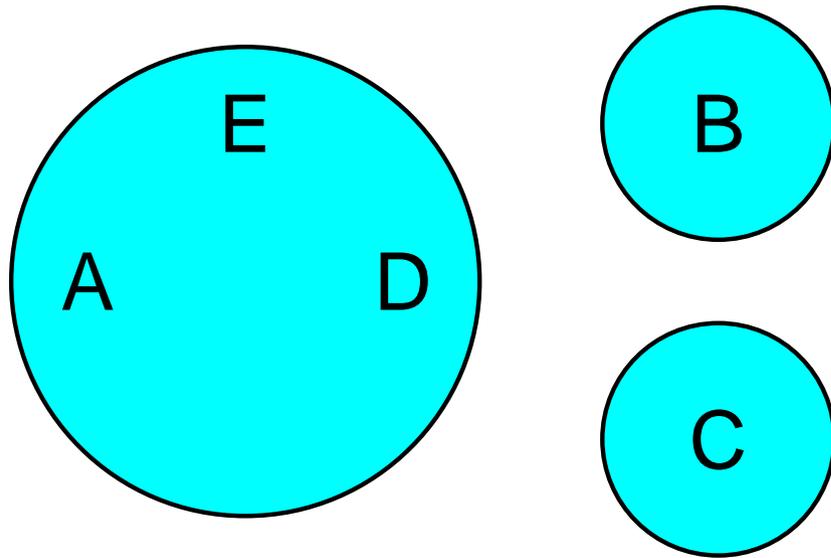


The union-find problem

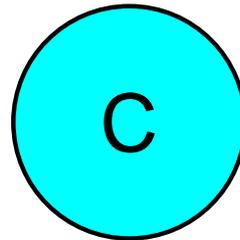
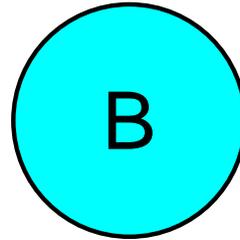
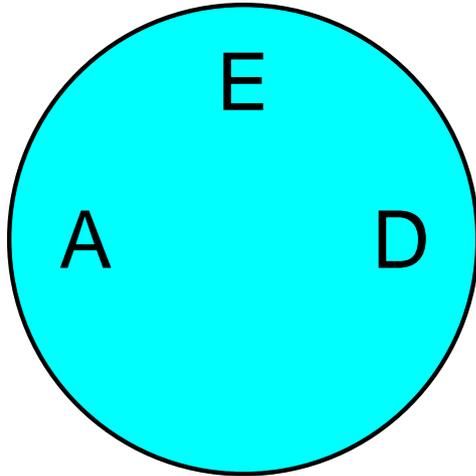
Bruce Merry



Background

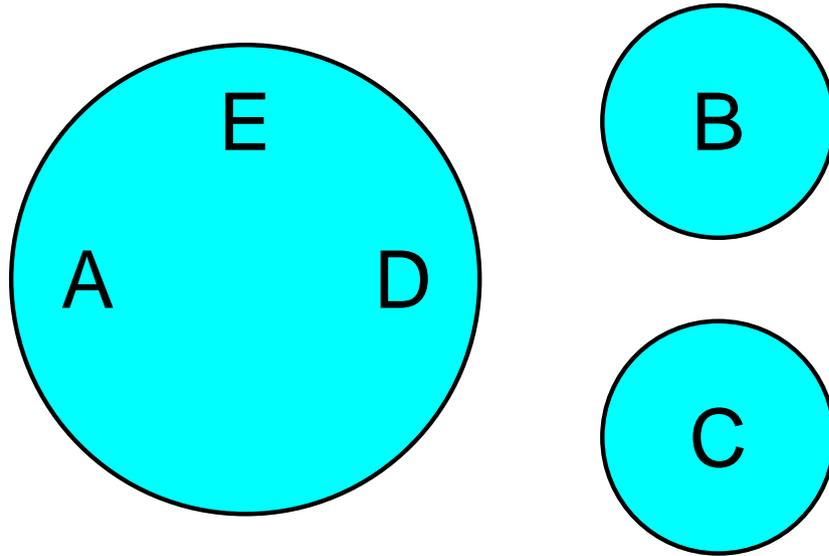


Simple solution 1



0	1	2	0	0
---	---	---	---	---

Simple solution 2

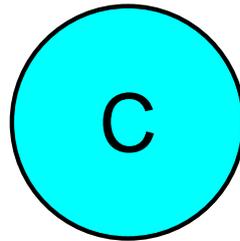
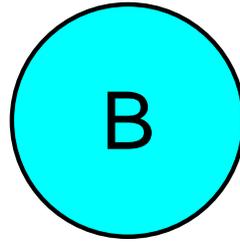
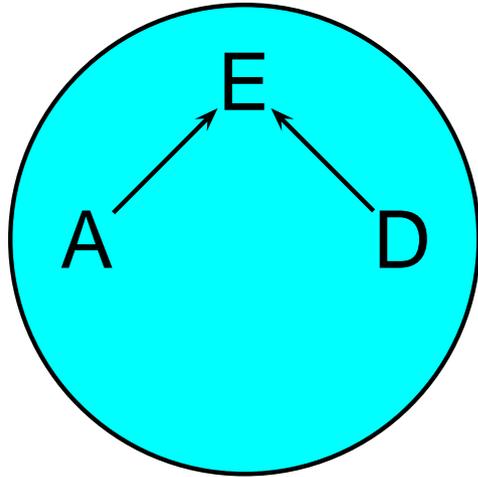


0: $A \rightarrow D \rightarrow E$

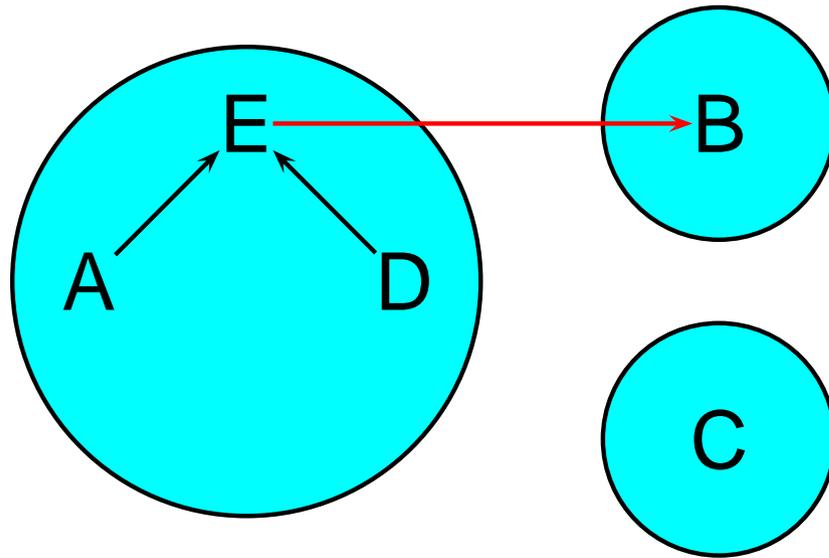
1: B

2: C

Trees



Trees



Pseudo-code

```
proc find( $x$ )  $\equiv$   
  while parent( $x$ )  $\geq 0$   
     $x := \textit{parent}(x)$   
  end
```

.

```
proc union( $x, y$ )  $\equiv$   
   $r_x := \textit{find}(x)$   
   $r_y := \textit{find}(y)$   
  if  $r_x \neq r_y$   
    parent( $r_y$ )  $:= r_x$   
  fi
```

.

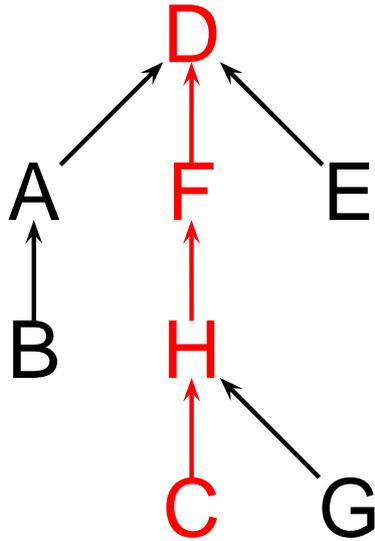
Balancing

Prevent degenerate trees:

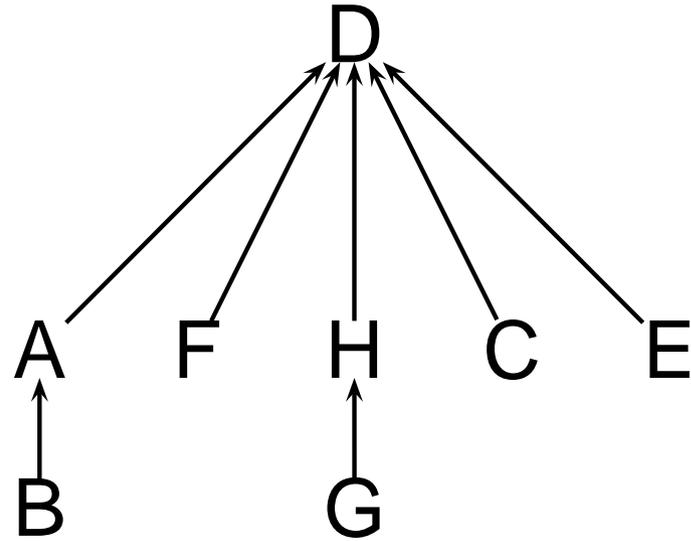
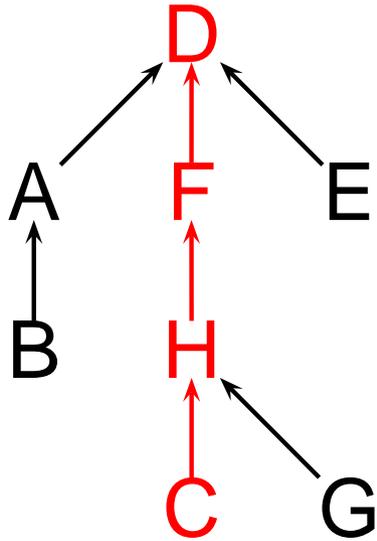
- Make shallower tree the child of the deeper
- Make smaller tree the child of the larger

Height/size can be stored in root.

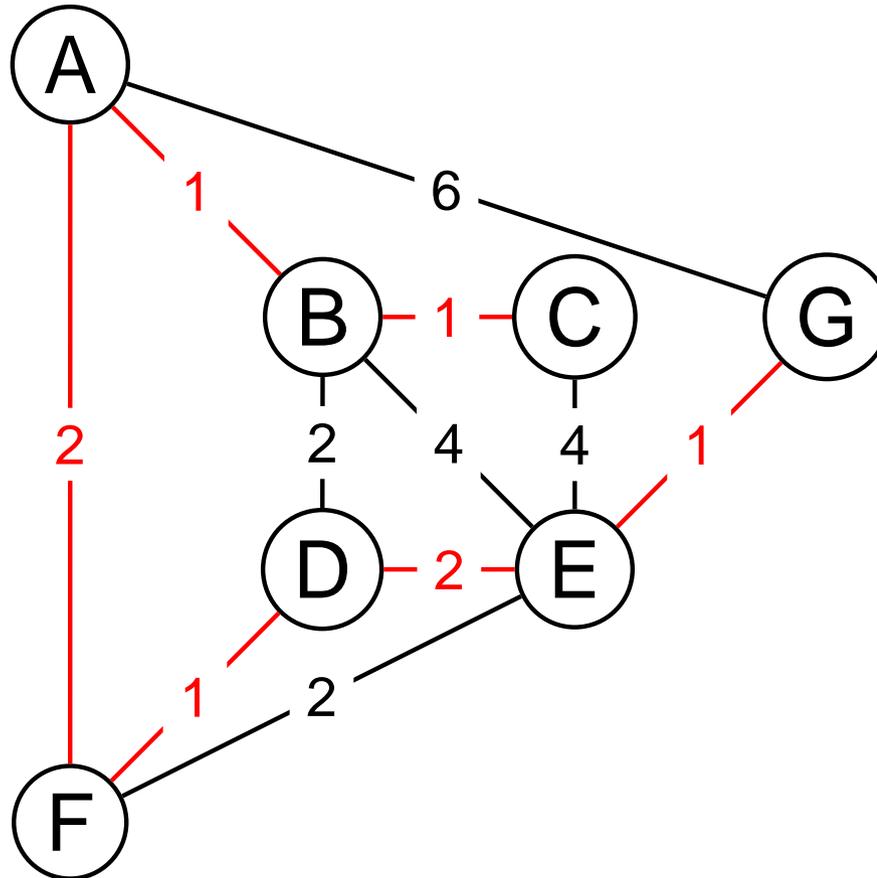
Path compression



Path compression



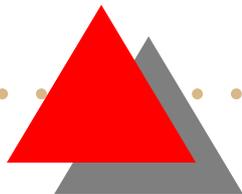
Minimum spanning trees





Kruskal's algorithm

- Start with no edges in the MST



Kruskal's algorithm

- Start with no edges in the MST
- Add the shortest unused edge that does not create a cycle



Kruskal's algorithm

- Start with no edges in the MST
- Add the shortest unused edge that does not create a cycle
- Repeat until $V - 1$ edges have been added.

