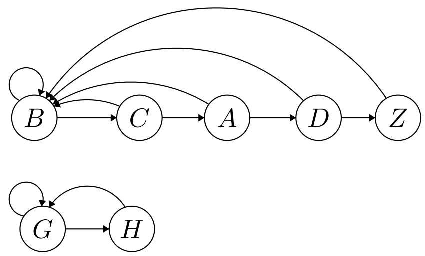
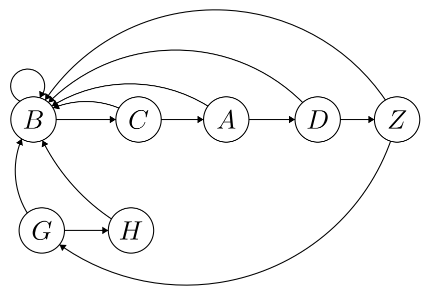
Lecture 22

Disjoint Set ADT

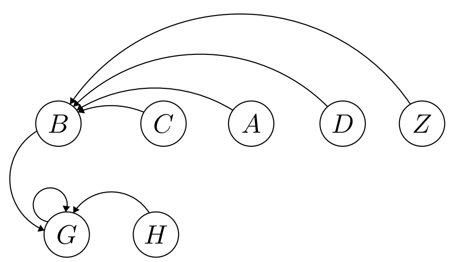
`Make-Set(`x`)`
`Find-Set(`x`)`
`Union(`x, y`)`



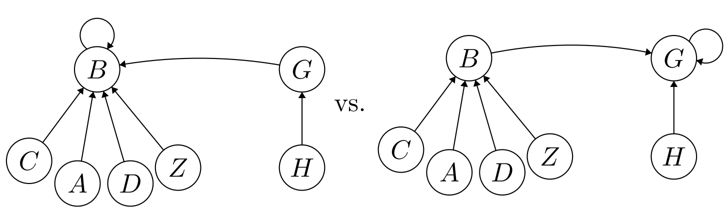
Union by weight:



Trees:

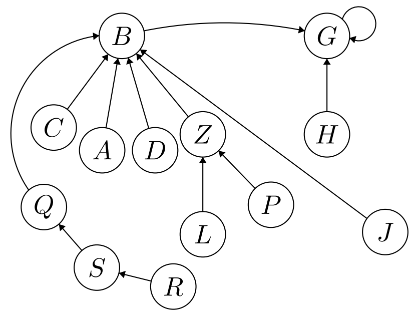


Union by weight with trees (larger tree becomes root - B *vs.* G):

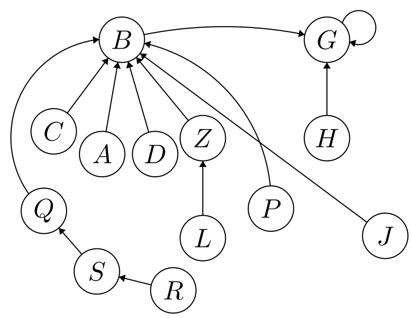


`if Find-Set(`u_i`) != Find-Set(`v_i`):`
  `union(`u_i, v_i`)`
  $T \leftarrow U \cup \{(u_i, v_i)\}$

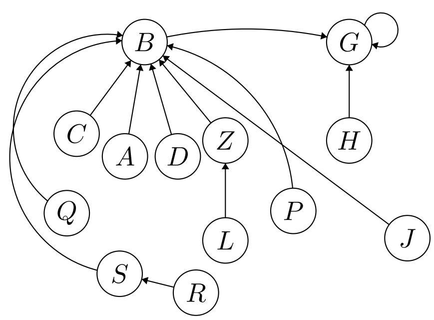
Path compression:



`Find-Set(`P`)`:



`Find-Set(`S`)`:



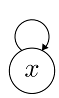
Union by Rank

- upper bound on height

$$h' < h \to h$$
$$h' = h \to h + 1$$

Trees with Union by Rank + Path Compression

`Make-Set(`x`)`: $\text{rank}_x = 0$,



`Union`(x, y): the root with higher rank becomes the new root and is unchanged. If two ranks are equal, either root is chosen as new root and rank is incremented.

`Find-Set`(x): use path compression and leave ranks unchanged

It is possible to prove that the worst-case running time for a sequence of m operations where there are are n `Make-Set` operations is $O(m \log *(n))$ or $O(m\alpha(n))$.

$$log * n = \begin{cases} 0 & 0 \leq n \leq 2 \\ 1 & n = 3 \\ 2 & 4 \leq n \leq 7 \\ 3 & 8 \leq n \leq 2047 \\ 4 & 2048 \leq n \leq i \text{ where } i > 10^{80} \end{cases}$$