Lecture 20

Minimum Spanning Trees (cont.)

Kruskal's MST Algorithm

`kruskal(`$G = (V, E)$`,` $w : E \to R$`):`
  $T \leftarrow \{\}$
  `sort edges so` $w(e_1) \leq w(e_2) \leq \ldots \leq w(e_n)$
  `for` $i \leftarrow 1$ `to` n
    `# let` $(u_i, v_i) = e_i$
    `if` u_i, v_i `in different connected components of` T`:`
      $T \leftarrow T \cup \{e_i\}$

Disjoint Set ADT

Objects: Collection of non-empty sets

$$s = \{s_1, s_2, \ldots, s_k\}$$

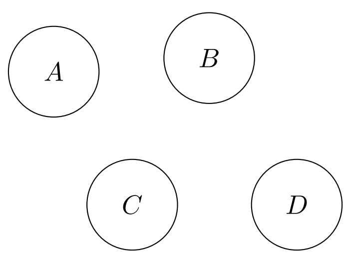
where each s_i is a non-empty set that has no element in common with any other s_i. That is, $s_i \cap s_j = \{\}$. Each set is identified by a unique element called its "representative".

Operations:

`make-set(`x`)`: Given element x that does not already belong to one of the sets, create a new set $\{x\}$ that contains only x (and assign x as the representative).

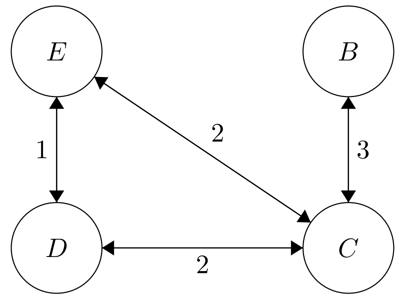
`find-set(`x`)`: Given element x, return the representative of the set that contains x (or NIL if x not in a set).

`union(`x, y`)`: Given two distinct elements x and y, let s_x be the set that contains x and s_y be the set that contains y. Form a new set consisting of $s_x \cup s_y$, and remove s_x and s_y from the collection. Pick a representative for the new set.



$$\{A\}, \{B\}, \{C\}, \{D\} \to \texttt{union}(A, B) \to \{A, \hat{B}\}, \{C\}, \{D\}.$$

`kruskal(`$G = (V, E)$`,` $w : E \to R$`):`
  $T \leftarrow \{\}$
  `sort edges so` $w(e_1) \leq w(e_2) \leq \ldots \leq w(e_n)$
  `for` $v \in V$`:`
    `makeset(`v`)`
  `for` $i \leftarrow 1$ `to` $|E|$
    `# let` $(u_i, v_i) = e_i$
    `if find-set(`u_i`) != find-set(`v_i`):`
      $T \leftarrow T \cup \{e_i\}$
    `if` u_i, v_i `in different connected components of` T`:`
      `union(`u_i, v_i`)`



$$\{\hat{C}\}, \{\hat{B}\}, \{\hat{D}\}, \{\hat{E}\} \rightarrow \{\hat{C}\}, \{\hat{B}\}, \{E, \hat{D}\} \rightarrow \{B\}, \{\hat{D}, C, E\} \rightarrow \{\hat{B}, D, C, E\}$$