CS711008Z Algorithm Design and Analysis

Lecture 7. KRUSKAL'S algorithm for MINIMUM SPANNING TREE [1]

Dongbo Bu

Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China

[1] The slides were made based on Chapter 5 of Algorithms, and Data Structure by Ellis Horowitz.

Kruskal's MST algorithm

Kruskal's algorithm [1956]

- Basic idea: during the execution, F is always an **acyclic forest**, and the **safe edge** added to F is always a least-weight edge connecting two distinct components.

Figure 1: Joseph Kruskal

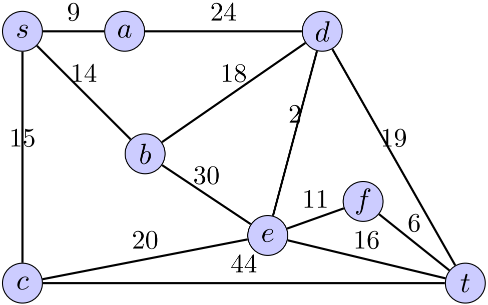
Kruskal's algorithm [1956]

MST-KRUSKAL(G, W)

$F = \{\}$;
for all vertex $v \in V$ **do**
    MAKESET(v);
end for
sort the edges of E into nondecreasing order by weight W;
for each edge $(u, v) \in E$ in the order **do**
    if FINDSET$(u) \neq$ FINDSET(v) **then**
        $F = F \cup \{(u, v)\}$;
        UNION (u, v);
    end if
end for

Here, UNION-FIND structure is used to detect whether a set of edges form a cycle.

Kruskal's MST algorithm: an example

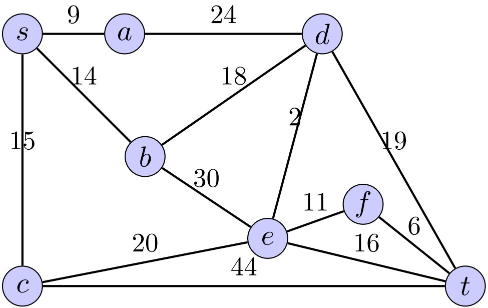


Kruskal's MST algorithm: an example

Step 1

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{s\}, \{t\}$

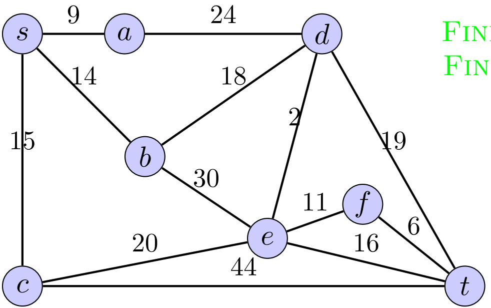


Kruskal's MST algorithm: an example

Step 1

Edge weight: 2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44

Disjoint sets: $\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{s\}, \{t\}$



$\textsc{Find}(d)$ returns $\{d\}$

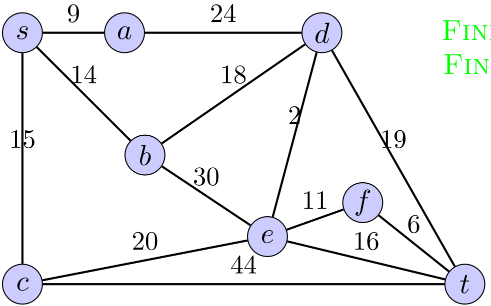
$\textsc{Find}(e)$ returns $\{e\}$

Kruskal's MST algorithm: an example

Step 1

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{s\}, \{t\}$



$\textsc{Find}(d)$ returns $\{d\}$

$\textsc{Find}(e)$ returns $\{e\}$

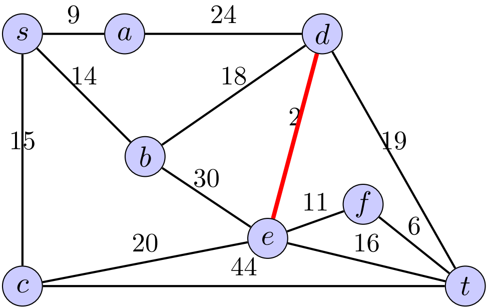
$\textsc{Union}(d, e)$

Kruskal's MST algorithm: an example

Step 1

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a\}, \{b\}, \{c\}, \{d, e\}, \{f\}, \{s\}, \{t\}$

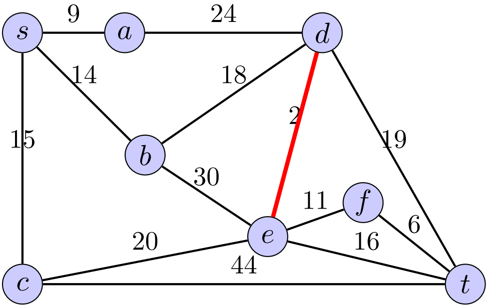


Kruskal's MST algorithm: an example

Step 2

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a\}, \{b\}, \{c\}, \{d, e\}, \{f\}, \{s\}, \{t\}$

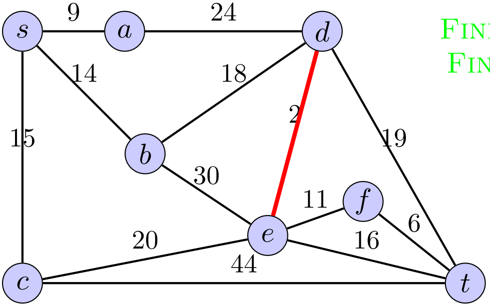


Kruskal's MST algorithm: an example

Step 2

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a\}, \{b\}, \{c\}, \{d, e\}, \{f\}, \{s\}, \{t\}$



$\text{FIND}(f)$ returns $\{f\}$

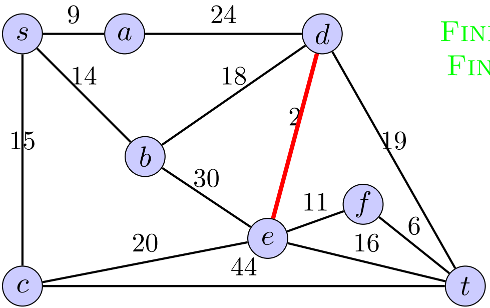
$\text{FIND}(t)$ returns $\{t\}$

Kruskal's MST algorithm: an example

Step 2

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a\}, \{b\}, \{c\}, \{d, e\}, \{f\}, \{s\}, \{t\}$



$\textsc{Find}(f)$ returns $\{f\}$

$\textsc{Find}(t)$ returns $\{t\}$

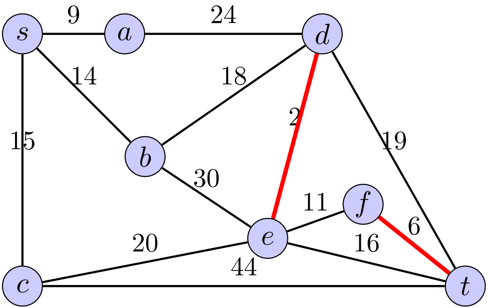
$\textsc{Union}(f, t)$

Kruskal's MST algorithm: an example

Step 2

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a\}, \{b\}, \{c\}, \{d, e\}, \{f, t\}, \{s\}$

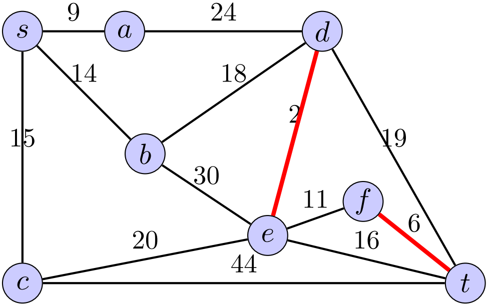


Kruskal's MST algorithm: an example

Step 3

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a\}, \{b\}, \{c\}, \{d, e\}, \{f, t\}, \{s\}$

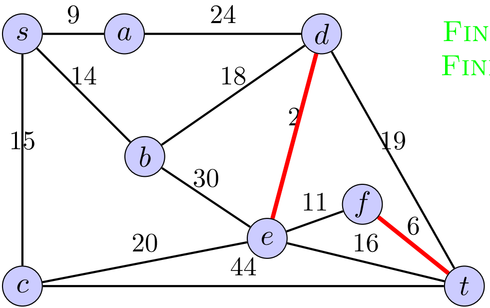


Kruskal's MST algorithm: an example

Step 3

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a\}, \{b\}, \{c\}, \{d, e\}, \{f, t\}, \{s\}$



$\textsc{Find}(s)$ returns $\{s\}$

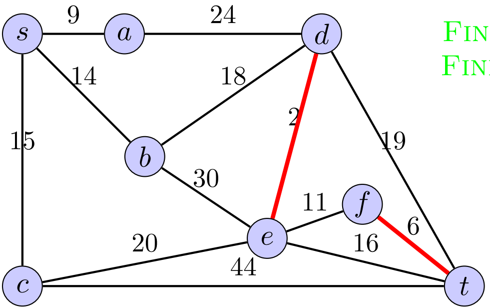
$\textsc{Find}(a)$ returns $\{a\}$

Kruskal's MST algorithm: an example

Step 3

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a\}, \{b\}, \{c\}, \{d, e\}, \{f, t\}, \{s\}$



$\textsc{Find}(s)$ returns $\{s\}$

$\textsc{Find}(a)$ returns $\{a\}$

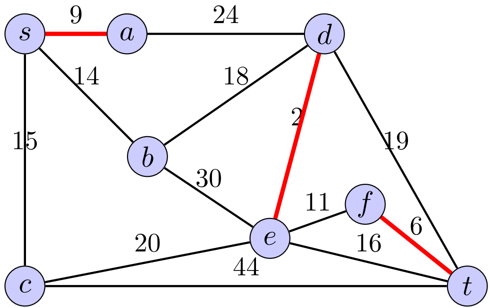
$\textsc{Union}(s, a)$

Kruskal's MST algorithm: an example

Step 3

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s\}, \{b\}, \{c\}, \{d, e\}, \{f, t\}$

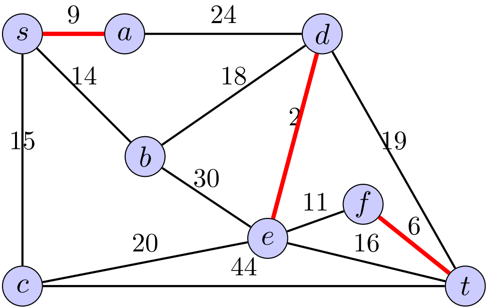


Kruskal's MST algorithm: an example

Step 4

Edge weight: $2, 6, 9, {\color{green}11}, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s\}, \{b\}, \{c\}, \{d, e\}, \{f, t\}$

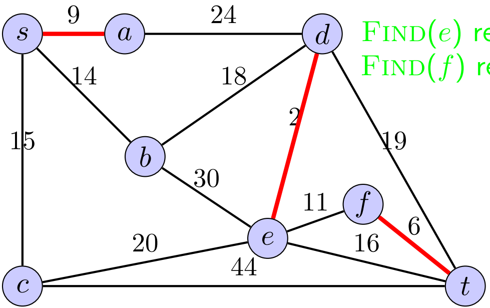


Kruskal's MST algorithm: an example

Step 4

Edge weight: $2, 6, 9, {\color{green}11}, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s\}, \{b\}, \{c\}, \{d, e\}, \{f, t\}$



$\textsc{Find}(e)$ returns $\{d, e\}$

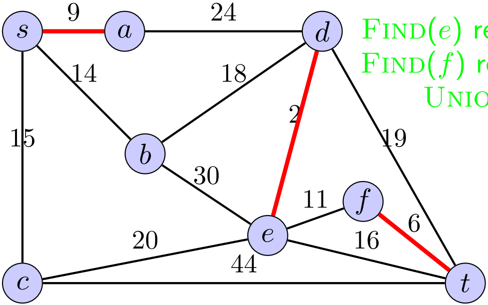
$\textsc{Find}(f)$ returns $\{f, t\}$

Kruskal's MST algorithm: an example

Step 4

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s\}, \{b\}, \{c\}, \{d, e\}, \{f, t\}$



$\textsc{Find}(e)$ returns $\{d, e\}$

$\textsc{Find}(f)$ returns $\{f, t\}$

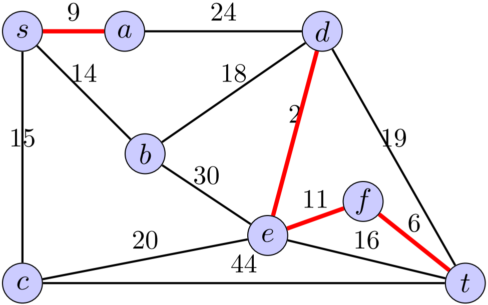
$\textsc{Union}(e, f)$

Kruskal's MST algorithm: an example

Step 4

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s\}, \{b\}, \{c\}, \{d, e, f, t\}$

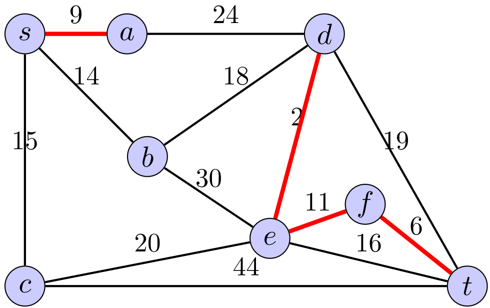


Kruskal's MST algorithm: an example

Step 5

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s\}, \{b\}, \{c\}, \{d, e, f, t\}$

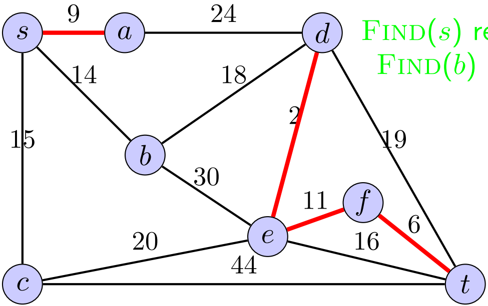


Kruskal's MST algorithm: an example

Step 5

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s\}, \{b\}, \{c\}, \{d, e, f, t\}$



$\textsc{Find}(s)$ returns $\{s, a\}$

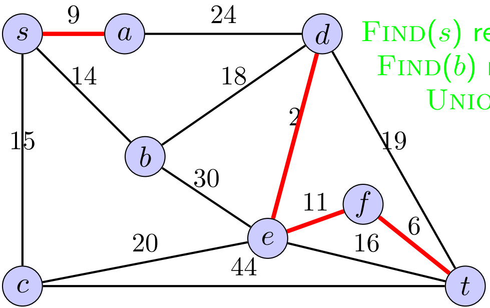
$\textsc{Find}(b)$ returns $\{b\}$

Kruskal's MST algorithm: an example

Step 5

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s\}, \{b\}, \{c\}, \{d, e, f, t\}$



$\textsc{Find}(s)$ returns $\{s, a\}$

$\textsc{Find}(b)$ returns $\{b\}$

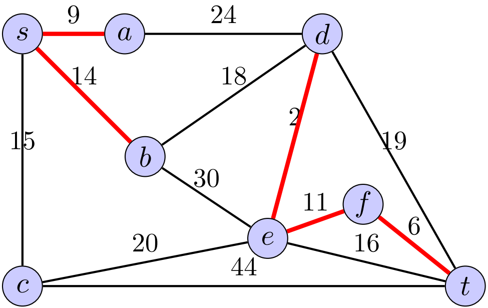
$\textsc{Union}(s, b)$

Kruskal's MST algorithm: an example

Step 5

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s, b\}, \{c\}, \{d, e, f, t\}$

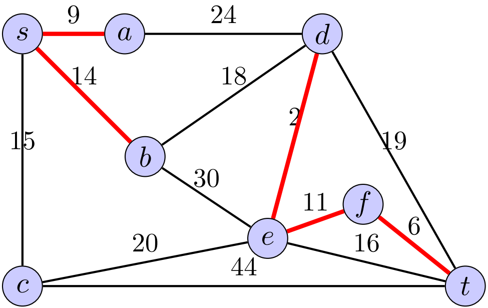


Kruskal's MST algorithm: an example

Step 6

Edge weight: $2, 6, 9, 11, 14, {\color{green}15}, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s, b\}, \{c\}, \{d, e, f, t\}$

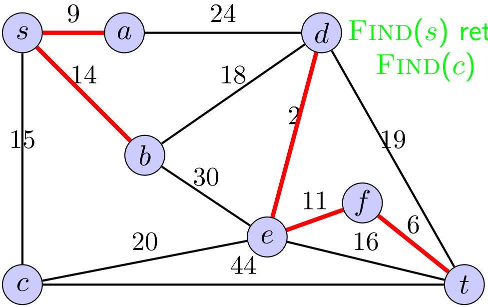


Kruskal's MST algorithm: an example

Step 6

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s, b\}, \{c\}, \{d, e, f, t\}$



$\textsc{Find}(s)$ returns $\{s, a, b\}$

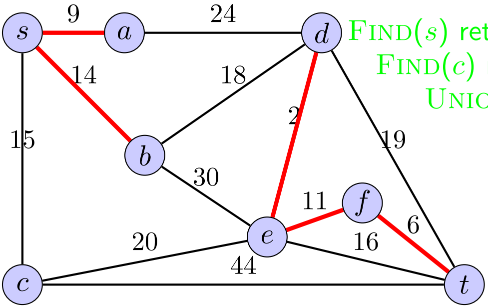
$\textsc{Find}(c)$ returns $\{c\}$

Kruskal's MST algorithm: an example

Step 6

Edge weight: $2, 6, 9, 11, 14, \textcolor{green}{15}, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s, b\}, \{c\}, \{d, e, f, t\}$



$\textsc{Find}(s)$ returns $\{s, a, b\}$

$\textsc{Find}(c)$ returns $\{c\}$

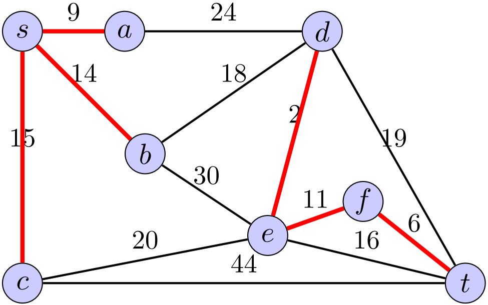
$\textsc{Union}(s, c)$

Kruskal's MST algorithm: an example

Step 6

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s, b, c\}, \{d, e, f, t\}$

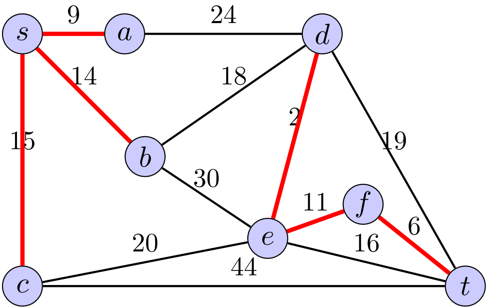


Kruskal's MST algorithm: an example

Step 7

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s, b, c\}, \{d, e, f, t\}$

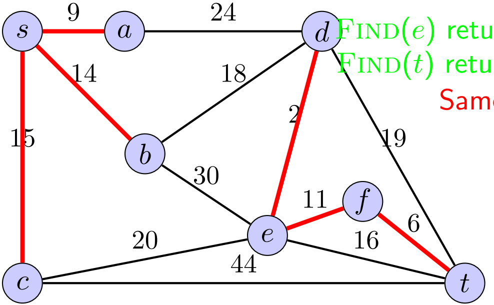


Kruskal's MST algorithm: an example

Step 7

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s, b, c\}, \{d, e, f, t\}$



$\text{FIND}(e)$ returns $\{d, e, f, t\}$

$\text{FIND}(t)$ returns $\{d, e, f, t\}$

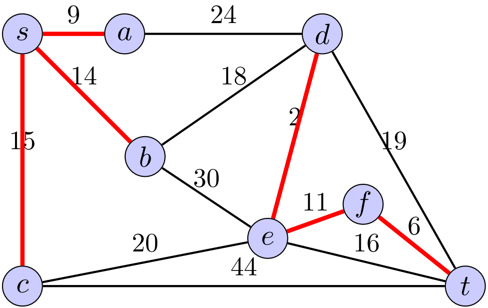
Same set!

Kruskal's MST algorithm: an example

Step 8

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s, b, c\}, \{d, e, f, t\}$

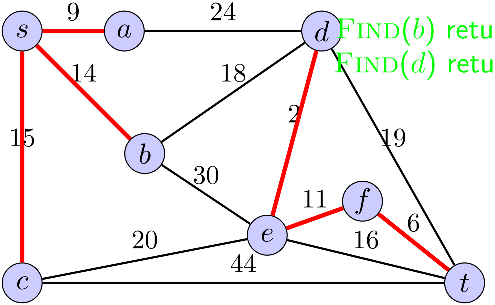


Kruskal's MST algorithm: an example

Step 8

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s, b, c\}, \{d, e, f, t\}$



$\textsc{Find}(b)$ returns $\{a, s, b, c\}$

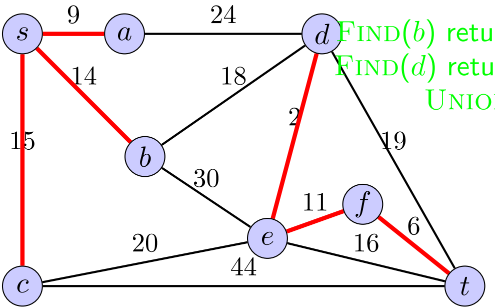
$\textsc{Find}(d)$ returns $\{d, e, f, t\}$

Kruskal's MST algorithm: an example

Step 8

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s, b, c\}, \{d, e, f, t\}$



$\textsc{Find}(b)$ returns $\{a, s, b, c\}$

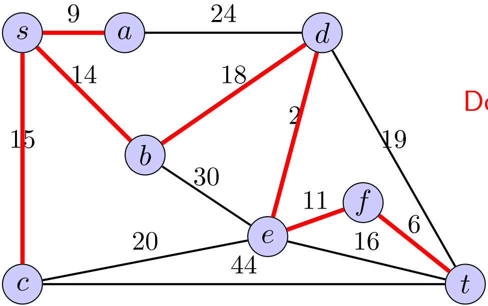
$\textsc{Find}(d)$ returns $\{d, e, f, t\}$

$\textsc{Union}(b, d)$

Step 8

Edge weight: $2, 6, 9, 11, 14, 15, 16, 18, 19, 20, 24, 30, 44$

Disjoint sets: $\{a, s, b, c, d, e, f, t\}$



Done!

Time complexity of KRUSKAL'S MST algorithm

Operation	Array	Tree	Link-by-size	Link-by-size + path compression
MAKESET	1	1	1	1
FIND	1	n	$\log n$	lg^*n
UNION	n	1	$\log n$	lg^*n
KRUSKAL'S MST	$O(n^2)$	$O(mn)$	$O(m \log n)$	$O(mlg^*n)$

KRUSKAL'S MST algorithm: n MAKESET, $n-1$ UNION, and m FIND.