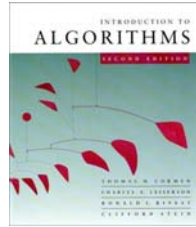**CS 5633 -- Spring 2008**

ALGORITHMS

Minimum Spanning Trees

Carola Wenk

Slides courtesy of Charles Leiserson with changes and additions by Carola Wenk

---

Minimum spanning trees

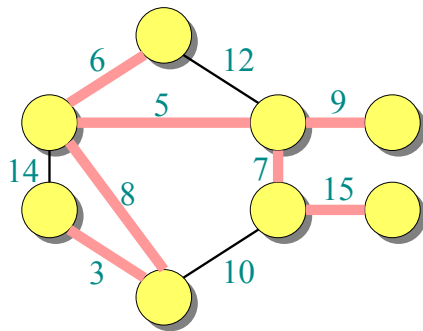**Input:** A connected, undirected graph $G = (V, E)$ with weight function $w : E \rightarrow \mathbf{R}$.
• For simplicity, assume that all edge weights are distinct. (CLRS covers the general case.)

Output: A *spanning tree* T — a tree that connects all vertices — of minimum weight:

$$w(T) = \sum_{(u,v) \in T} w(u,v).$$

Example of MST

Hallmark for "greedy" algorithms

Greedy-choice property
A locally optimal choice is globally optimal.

Theorem. Let T be the MST of $G = (V, E)$, and let $A \subseteq V$. Suppose that $(u, v) \in E$ is the least-weight edge connecting A to $V \setminus A$. Then, $(u, v) \in T$.
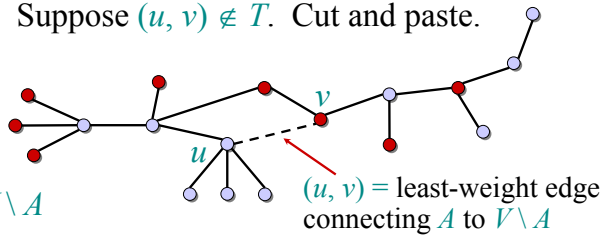
Proof of theorem

Proof. Suppose $(u, v) \notin T$. Cut and paste.

T:

○ $\in A$

● $\in V \setminus A$

v

u

(u, v) = least-weight edge connecting A to $V \setminus A$

Proof of theorem

Proof. Suppose $(u, v) \notin T$. Cut and paste.

T:

○ $\in A$

● $\in V \setminus A$

v

u

(u, v) = least-weight edge connecting A to $V \setminus A$

Consider the unique simple path from u to v in T.

Proof of theorem

Proof. Suppose $(u, v) \notin T$. Cut and paste.

T:

○ $\in A$

● $\in V \setminus A$

v

u

(u, v) = least-weight edge connecting A to $V \setminus A$

Consider the unique simple path from u to v in T.

Swap (u, v) with the first edge on this path that connects a vertex in A to a vertex in $V \setminus A$.

Proof of theorem

Proof. Suppose $(u, v) \notin T$. Cut and paste.

T':

○ $\in A$

● $\in V \setminus A$

v

u

(u, v) = least-weight edge connecting A to $V \setminus A$
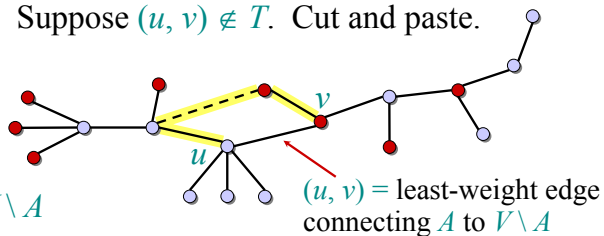
Consider the unique simple path from u to v in T.

Swap (u, v) with the first edge on this path that connects a vertex in A to a vertex in $V \setminus A$.
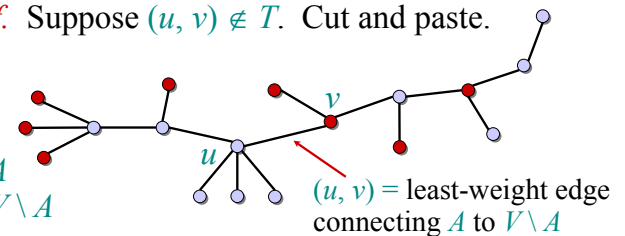
A lighter-weight spanning tree than T results. ☐

Slide 9

Prim's algorithm

IDEA: Maintain $V \setminus A$ as a priority queue Q. Key each vertex in Q with the weight of the least-weight edge connecting it to a vertex in A.

$Q \leftarrow V$
$key[v] \leftarrow \infty$ for all $v \in V$
$key[s] \leftarrow 0$ for some arbitrary $s \in V$
while $Q \neq \varnothing$
    do $u \leftarrow$ EXTRACT-MIN(Q)
        for each $v \in Adj[u]$
            do if $v \in Q$ and $w(u, v) < key[v]$
                then $key[v] \leftarrow w(u, v)$ ▷ DECREASE-KEY
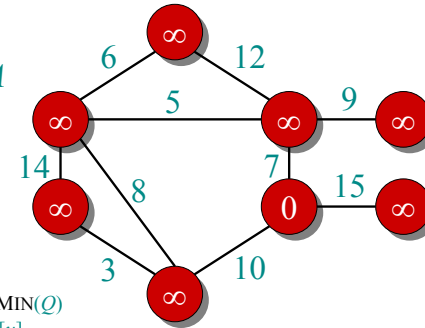 $\pi[v] \leftarrow u$

At the end, $\{(v, \pi[v])\}$ forms the MST.

Slide 10

Example of Prim's algorithm

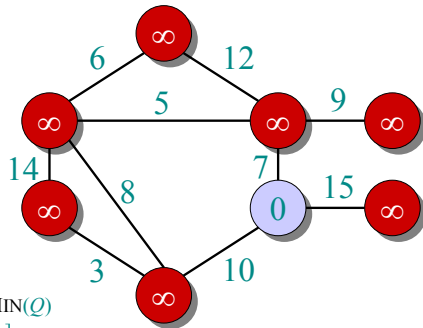$\circ \in A$
$\bullet \in V \setminus A$

$u \leftarrow$ EXTRACT-MIN(Q)
for each $v \in Adj[u]$
    do if $v \in Q$ and $w(u, v) < key[v]$
        then $key[v] \leftarrow w(u, v)$▷ DECREASE-KEY
        $\pi[v] \leftarrow u$

Slide 11

Example of Prim's algorithm

$\circ \in A$
$\bullet \in V \setminus A$

$u \leftarrow$ EXTRACT-MIN(Q)
for each $v \in Adj[u]$
    do if $v \in Q$ and $w(u, v) < key[v]$
        then $key[v] \leftarrow w(u, v)$▷ DECREASE-KEY
        $\pi[v] \leftarrow u$

Slide 12

Example of Prim's algorithm

$\circ \in A$
$\bullet \in V \setminus A$

$u \leftarrow$ EXTRACT-MIN(Q)
for each $v \in Adj[u]$
    do if $v \in Q$ and $w(u, v) < key[v]$
        then $key[v] \leftarrow w(u, v)$▷ DECREASE-KEY
        $\pi[v] \leftarrow u$

Example of Prim's algorithm

○ ∈ *A*
● ∈ *V* \ *A*

∞   6   12
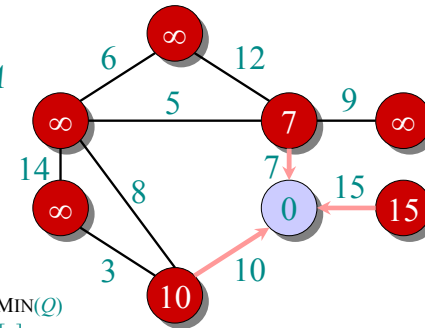∞   5   7   9   ∞
14   7
∞   8   0   15   15
3   10   10

$u \leftarrow$ EXTRACT-MIN(Q)
for each $v \in Adj[u]$
   do if $v \in Q$ and $w(u, v) < key[v]$
      then $key[v] \leftarrow w(u, v)$ ▷ DECREASE-KEY
        $\pi[v] \leftarrow u$

4/3/08   *CS 5633 Analysis of Algorithms*   13

Example of Prim's algorithm
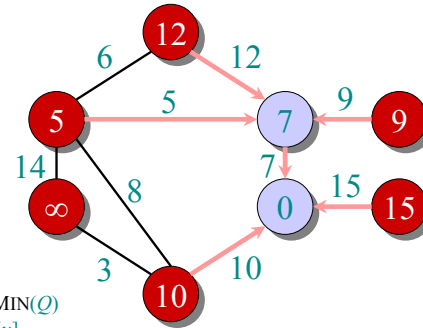
○ ∈ *A*
● ∈ *V* \ *A*

12   6   12
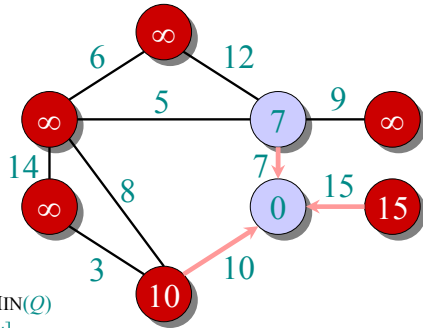5   5   7   9   9
14   7
∞   8   0   15   15
3   10   10

$u \leftarrow$ EXTRACT-MIN(Q)
for each $v \in Adj[u]$
   do if $v \in Q$ and $w(u, v) < key[v]$
      then $key[v] \leftarrow w(u, v)$ ▷ DECREASE-KEY
        $\pi[v] \leftarrow u$

4/3/08   *CS 5633 Analysis of Algorithms*   14

Example of Prim's algorithm
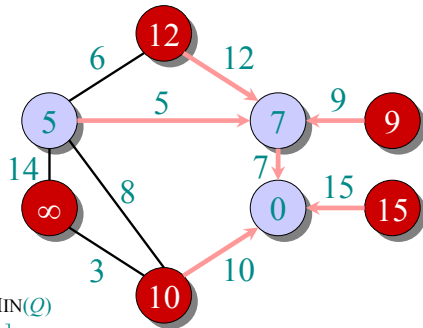
○ ∈ *A*
● ∈ *V* \ *A*

12   6   12
5   5   7   9   9
14   7
∞   8   0   15   15
3   10   10

$u \leftarrow$ EXTRACT-MIN(Q)
for each $v \in Adj[u]$
   do if $v \in Q$ and $w(u, v) < key[v]$
      then $key[v] \leftarrow w(u, v)$ ▷ DECREASE-KEY
        $\pi[v] \leftarrow u$

4/3/08   *CS 5633 Analysis of Algorithms*   15

Example of Prim's algorithm
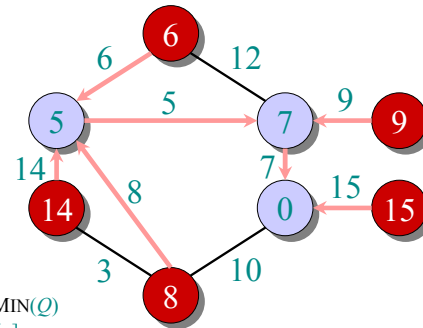
○ ∈ *A*
● ∈ *V* \ *A*

6   6   12
5   5   7   9   9
14   7
14   8   0   15   15
3   8   10

$u \leftarrow$ EXTRACT-MIN(Q)
for each $v \in Adj[u]$
   do if $v \in Q$ and $w(u, v) < key[v]$
      then $key[v] \leftarrow w(u, v)$ ▷ DECREASE-KEY
        $\pi[v] \leftarrow u$

4/3/08   *CS 5633 Analysis of Algorithms*   16

Example of Prim's algorithm
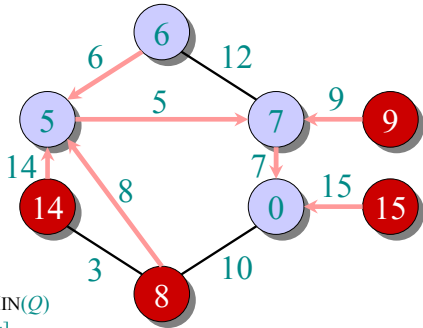
$\circ \in A$
$\bullet \in V \setminus A$

$u \leftarrow$ EXTRACT-MIN(Q)
for each $v \in Adj[u]$
    do if $v \in Q$ and $w(u, v) < key[v]$
        then $key[v] \leftarrow w(u, v)$ ▷ DECREASE-KEY
            $\pi[v] \leftarrow u$

4/3/08   *CS 5633 Analysis of Algorithms*   17

Example of Prim's algorithm
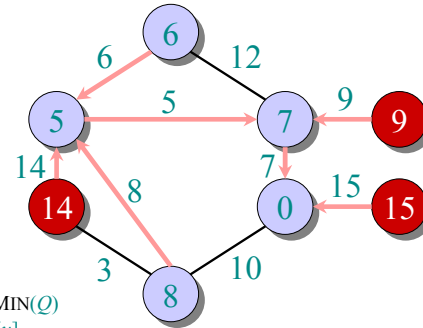
$\circ \in A$
$\bullet \in V \setminus A$

$u \leftarrow$ EXTRACT-MIN(Q)
for each $v \in Adj[u]$
    do if $v \in Q$ and $w(u, v) < key[v]$
        then $key[v] \leftarrow w(u, v)$ ▷ DECREASE-KEY
            $\pi[v] \leftarrow u$

4/3/08   *CS 5633 Analysis of Algorithms*   18

Example of Prim's algorithm
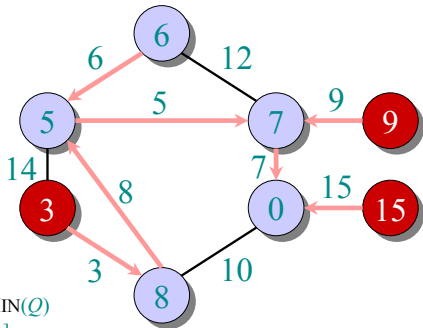
$\circ \in A$
$\bullet \in V \setminus A$

$u \leftarrow$ EXTRACT-MIN(Q)
for each $v \in Adj[u]$
    do if $v \in Q$ and $w(u, v) < key[v]$
        then $key[v] \leftarrow w(u, v)$ ▷ DECREASE-KEY
            $\pi[v] \leftarrow u$

4/3/08   *CS 5633 Analysis of Algorithms*   19

Example of Prim's algorithm
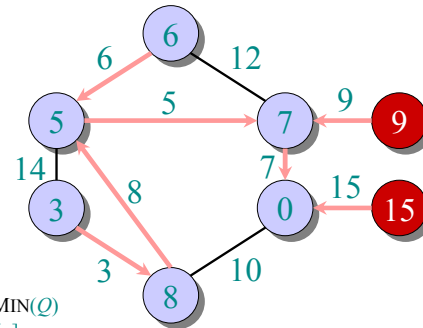
$\circ \in A$
$\bullet \in V \setminus A$

$u \leftarrow$ EXTRACT-MIN(Q)
for each $v \in Adj[u]$
    do if $v \in Q$ and $w(u, v) < key[v]$
        then $key[v] \leftarrow w(u, v)$ ▷ DECREASE-KEY
            $\pi[v] \leftarrow u$

4/3/08   *CS 5633 Analysis of Algorithms*   20

Example of Prim's algorithm

- $\circ \in A$
- $\bullet \in V \setminus A$

$$u \leftarrow \text{EXTRACT-MIN}(Q)$$
for each $v \in Adj[u]$
    do if $v \in Q$ and $w(u, v) < key[v]$
        then $key[v] \leftarrow w(u, v) \triangleright$ DECREASE-KEY
            $\pi[v] \leftarrow u$

4/3/08   *CS 5633 Analysis of Algorithms*   21

Example of Prim's algorithm

- $\circ \in A$
- $\bullet \in V \setminus A$

$$u \leftarrow \text{EXTRACT-MIN}(Q)$$
for each $v \in Adj[u]$
    do if $v \in Q$ and $w(u, v) < key[v]$
        then $key[v] \leftarrow w(u, v) \triangleright$ DECREASE-KEY
            $\pi[v] \leftarrow u$

4/3/08   *CS 5633 Analysis of Algorithms*   22
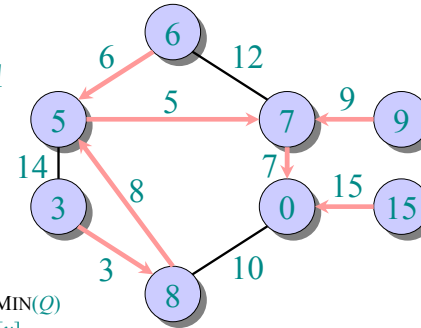
Analysis of Prim

$\Theta(|V|)$ total $\begin{cases} Q \leftarrow V \\ key[v] \leftarrow \infty \text{ for all } v \in V \\ key[s] \leftarrow 0 \text{ for some arbitrary } s \in V \end{cases}$

$|V|$ times $\begin{cases} \textbf{while } Q \neq \varnothing \\ \quad \textbf{do } u \leftarrow \text{EXTRACT-MIN}(Q) \\ \quad degree(u) \text{ times} \begin{cases} \textbf{for } \text{each } v \in Adj[u] \\ \quad \textbf{do if } v \in Q \text{ and } w(u, v) < key[v] \\ \quad\quad \textbf{then } key[v] \leftarrow w(u, v) \\ \quad\quad\quad \pi[v] \leftarrow u \end{cases} \end{cases}$

Handshaking Lemma $\Rightarrow \Theta(|E|)$ implicit DECREASE-KEY's.

Time $= \Theta(|V|) \cdot T_{\text{EXTRACT-MIN}} + \Theta(|E|) \cdot T_{\text{DECREASE-KEY}}$

4/3/08   *CS 5633 Analysis of Algorithms*   23

Analysis of Prim (continued)

Time $= \Theta(|V|) \cdot T_{\text{EXTRACT-MIN}} + \Theta(|E|) \cdot T_{\text{DECREASE-KEY}}$

Q	$T_{\text{EXTRACT-MIN}}$	$T_{\text{DECREASE-KEY}}$	Total
array	$O(|V|)$	$O(1)$	$O(|V|^2)$
binary heap	$O(\log|V|)$	$O(\log|V|)$	$O(|E| \log|V|)$
Fibonacci heap	$O(\log|V|)$ amortized	$O(1)$ amortized	$O(|E| + |V| \log|V|)$ worst case

4/3/08   *CS 5633 Analysis of Algorithms*   24
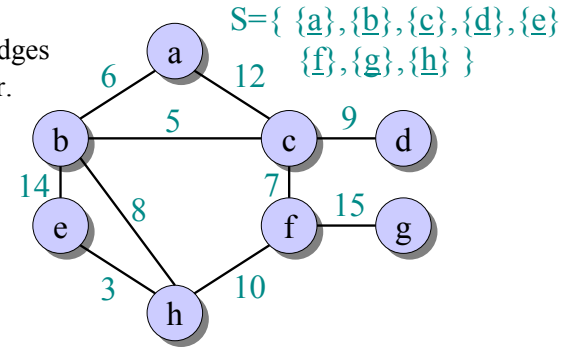
Kruskal's algorithm

IDEA (again greedy):
Repeatedly pick edge with smallest weight as long as it does not form a cycle.

• The algorithm creates a set of trees (a **forest**)
• During the algorithm the added edges merge the trees together, such that in the end only one tree remains

• The correctness of this greedy strategy is not obvious and needs to be proven. (Proof skipped here.)

4/3/08   *CS 5633 Analysis of Algorithms*   25

Example of Kruskal's algorithm

— MST edges
<u>a</u>   set repr.

S={ {<u>a</u>},{<u>b</u>},{<u>c</u>},{<u>d</u>},{<u>e</u>} {<u>f</u>},{<u>g</u>},{<u>h</u>} }

Every node is a single tree.

4/3/08   *CS 5633 Analysis of Algorithms*   26

Example of Kruskal's algorithm

— MST edges
<u>a</u>   set repr.

S={ {<u>a</u>},{<u>b</u>},{<u>c</u>},{<u>d</u>},{<u>f</u>} {<u>g</u>}, {<u>e</u>, h} }
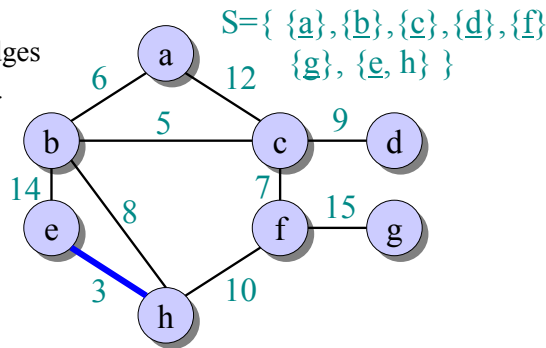
Edge 3 merged two singleton trees.
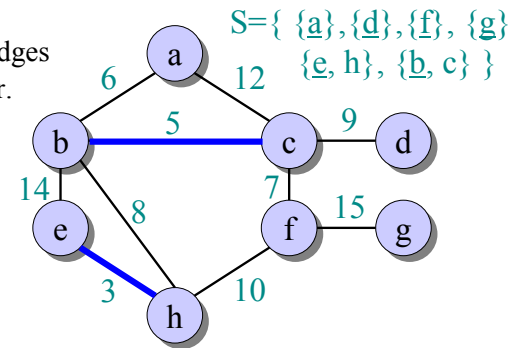
4/3/08   *CS 5633 Analysis of Algorithms*   27

Example of Kruskal's algorithm

— MST edges
<u>a</u>   set repr.

S={ {<u>a</u>},{<u>d</u>},{<u>f</u>}, {<u>g</u>} {<u>e</u>, h}, {<u>b</u>, c} }

4/3/08   *CS 5633 Analysis of Algorithms*   28

Example of Kruskal's algorithm

— MST edges
a̲ set repr.

S={ {d̲},{f̲}, {g̲}
{e̲, h}, {a̲, b, c} }

a
6   12
b   5   c   9   d
14   7
e   8   f   15   g
3   h   10

Example of Kruskal's algorithm

— MST edges
a̲ set repr.

S={ {d̲}, {g̲}
{e̲, h}, {a̲, b, c, f} }

a
6   12
b   5   c   9   d
14   7
e   8   f   15   g
3   h   10

Example of Kruskal's algorithm

— MST edges
a̲ set repr.

S={ {d̲}, {g̲}
{e̲, h, a, b, c, f} }

a
6   12
b   5   c   9   d
14   7
e   8   f   15   g
3   h   10
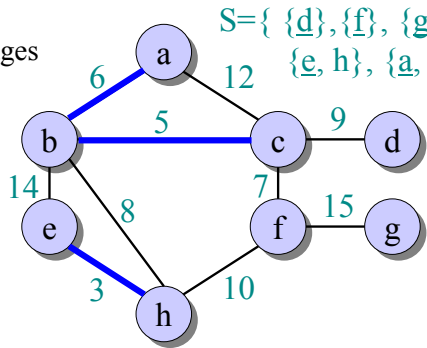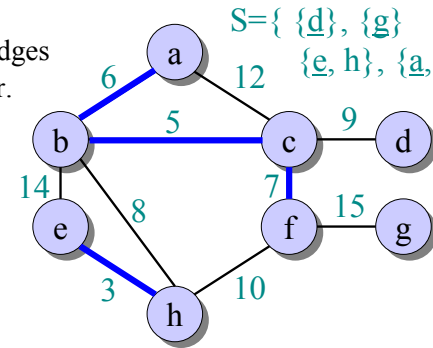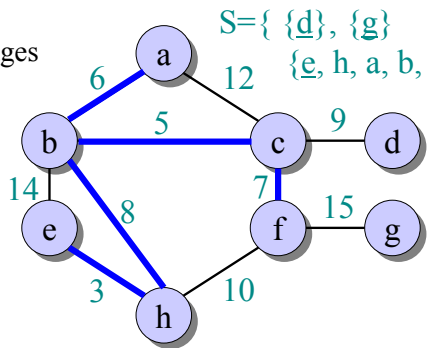
Edge 8 merged the two bigger trees.

Example of Kruskal's algorithm

— MST edges
a̲ set repr.

S={ {g̲}
{e̲, h, a, b, c, f, d} }

a
6   12
b   5   c   9   d
14   7
e   8   f   15   g
3   h   10

Slide 33

Example of Kruskal's algorithm

— MST edges

a set repr.

S={ {g}
{e, h, a, b, c, f, d} }

a
6   12
b   5   c   9   d
14   7   15
e   8   f   g
3   h   10
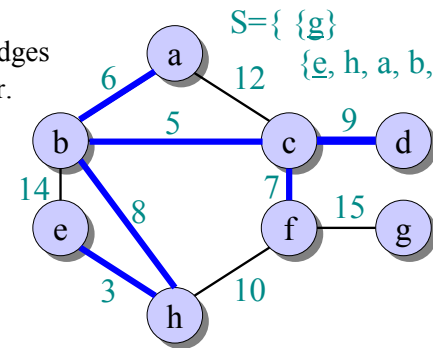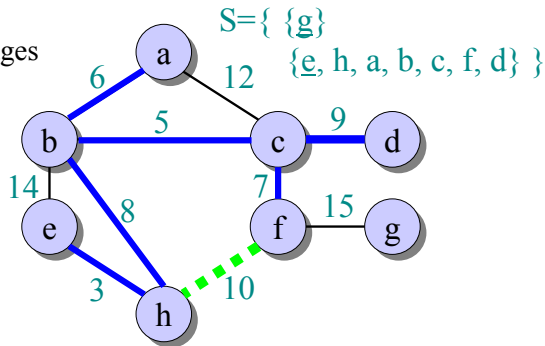
Skip edge 10 as it would cause a cycle.

4/3/08   *CS 5633 Analysis of Algorithms*   33

Slide 34

ALGORITHMS

Example of Kruskal's algorithm

— MST edges

a set repr.

S={ {g}
{e, h, a, b, c, f, d} }

a
6   12
b   5   c   9   d
14   7   15
e   8   f   g
3   h   10
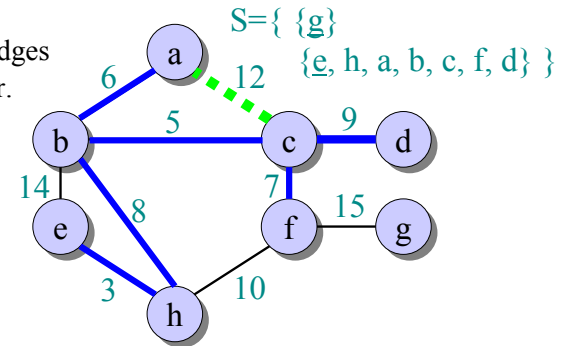
Skip edge 12 as it would cause a cycle.

4/3/08   *CS 5633 Analysis of Algorithms*   34

Slide 35

ALGORITHMS

Example of Kruskal's algorithm

— MST edges

a set repr.

S={ {g}
{e, h, a, b, c, f, d} }

a
6   12
b   5   c   9   d
14   7   15
e   8   f   g
3   h   10
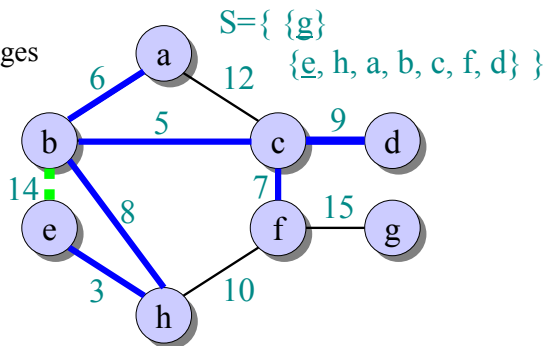
Skip edge 14 as it would cause a cycle.

4/3/08   *CS 5633 Analysis of Algorithms*   35

Slide 36

ALGORITHMS

Example of Kruskal's algorithm

— MST edges

a set repr.

S={{e, h, a, b, c, f, d, g} }

a
6   12
b   5   c   9   d
14   7   15
e   8   f   g
3   h   10
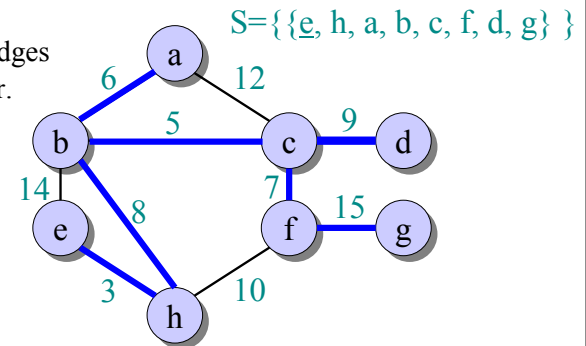
4/3/08   *CS 5633 Analysis of Algorithms*   36

Disjoint-set data structure (Union-Find)

- Maintains a dynamic collection of *pairwise-disjoint* sets $S = \{S_1, S_2, \ldots, S_r\}$.
- Each set S_i has one element distinguished as the **representative** element.
- Supports operations:

  $O(1)$ • MAKE-SET(x): adds new set $\{x\}$ to S

  $O(\alpha(n))$ • UNION(x, y): replaces sets S_x, S_y with $S_x \cup S_y$

  $O(\alpha(n))$ • FIND-SET(x): returns the representative of the set S_x containing element x

- $1 < \alpha(n) < \log^*(n) < \log(\log(n)) < \log(n)$

Kruskal's algorithm

IDEA: Repeatedly pick edge with smallest weight as long as it does not form a cycle.

$S \leftarrow \varnothing$ ▷ *S* will contain all MST edges

$O(|V|)$ **for** each $v \in V$ **do** MAKE-SET(v)

$O(|E|\log|E|)$ Sort edges of E in non-decreasing order according to w

$O(|E|)$ **For** each $(u,v) \in E$ taken in this order **do**

$O(\alpha(|V|))$ $\begin{cases} \textbf{if } \text{FIND-SET}(u) \neq \text{FIND-SET}(v) \text{ ▷ } u,v \text{ in different trees} \\ \quad A \leftarrow A \cup \{(u,v)\} \\ \quad \text{UNION}(u,v) \text{ ▷ Edge } (u,v) \text{ connects the two trees} \end{cases}$

Runtime: $O(|V|+|E|\log|E|+|E|\alpha(|V|)) = O(|E| \log |E|)$

MST algorithms

- Prim's algorithm:
  - Maintains one tree
  - Runs in time $O(|E| \log |V|)$, with binary heaps.
- Kruskal's algorithm:
  - Maintains a forest and uses the disjoint-set data structure
  - Runs in time $O(|E| \log |E|)$
- Best to date: Randomized algorithm by Karger, Klein, Tarjan [1993]. Runs in expected time $O(|V| + |E|)$