Lecture 21

CSE 331

Dijkstra's shortest path algorithm

$d'(w) = \min_{e=(u,w) \text{ in } E,\ u \text{ in } R} d(u)+l_e$

Input: Directed $G=(V,E)$, $l_e \geq 0$, s in V

$R = \{s\}$, $d(s) = 0$

While there is a x not in R with (u,x) in E, u in R

Pick w that minimizes $d'(w)$

Add w to R

$d(w) = d'(w)$

At most n iterations

$\Sigma_{x \in V} O(In_x+1)$
$= O(m+n)$ time

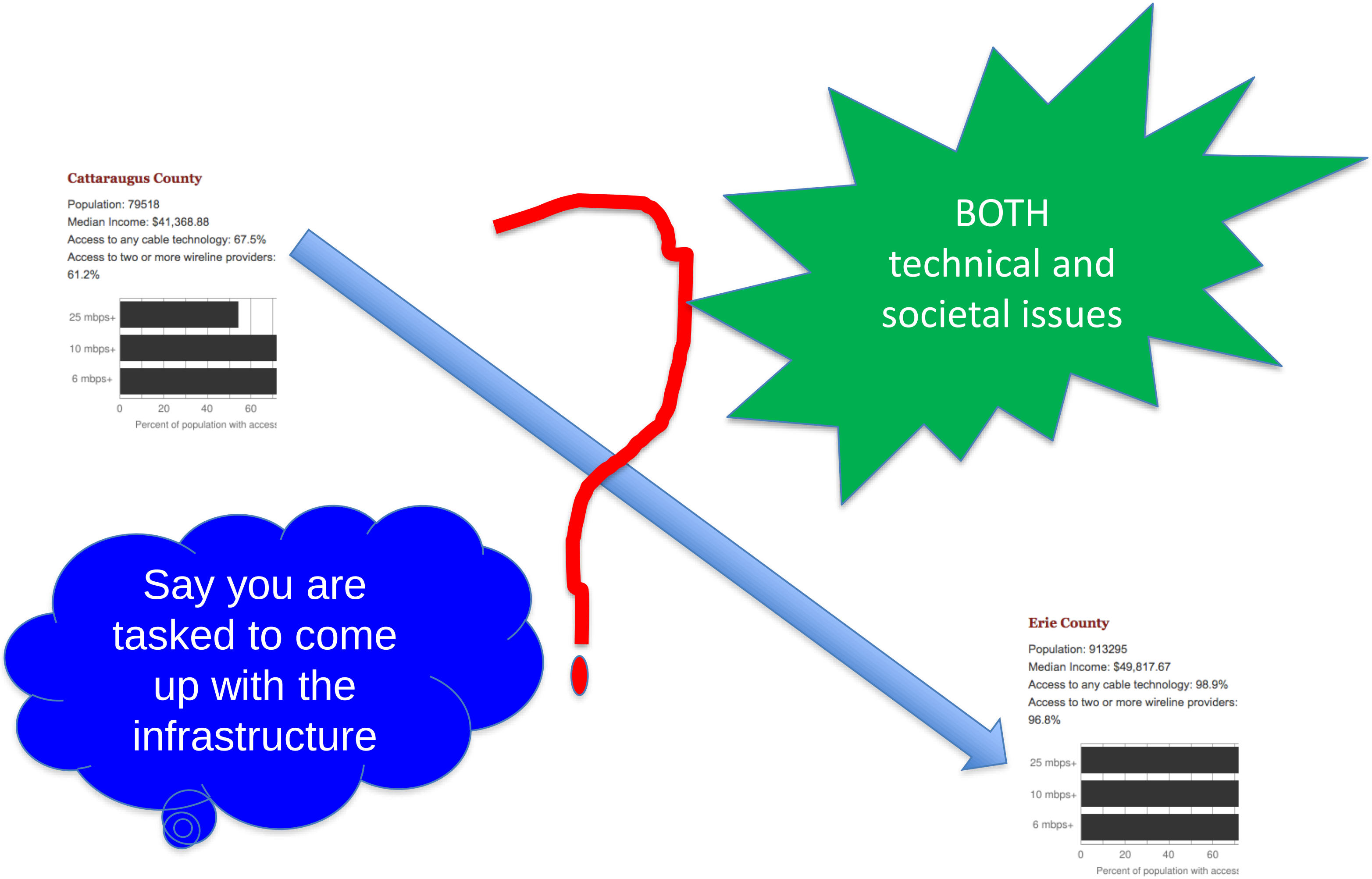
$O((m+n)n)$ time bound is trivial

$O((m+n) \log n)$ time implementation with priority Q

Reading Assignment

Sec 4.4 of [KT]

Make broadband more available



Building a fiber network

Lay down fibers to connect n locations

All n locations should be connected

Laying down a fiber costs money

What is the cheapest way to lay down the fibers?

Today's agenda

Minimum Spanning Tree (MST) Problem

Greedy algorithm(s) for MST problem

On to the board…

Minimum Spanning Tree Problem

Input: Undirected, connected $G = (V,E)$, edge costs c_e

Output: Subset $E' \subseteq E)$, s.t. $T = (V,E')$ is connected
$C(T)$ is minimized

If all $c_e > 0$, then T is indeed a tree

Rest of today’s agenda

Greedy algorithm(s) for MST problem

Kruskal's Algorithm

Joseph B. Kruskal

Input: $G=(V,E)$, $c_e > 0$ for every e in E

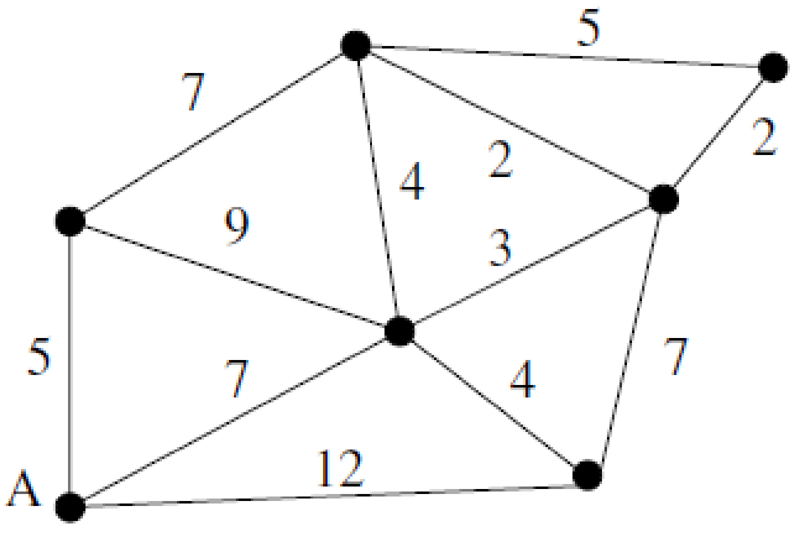
$T = \emptyset$

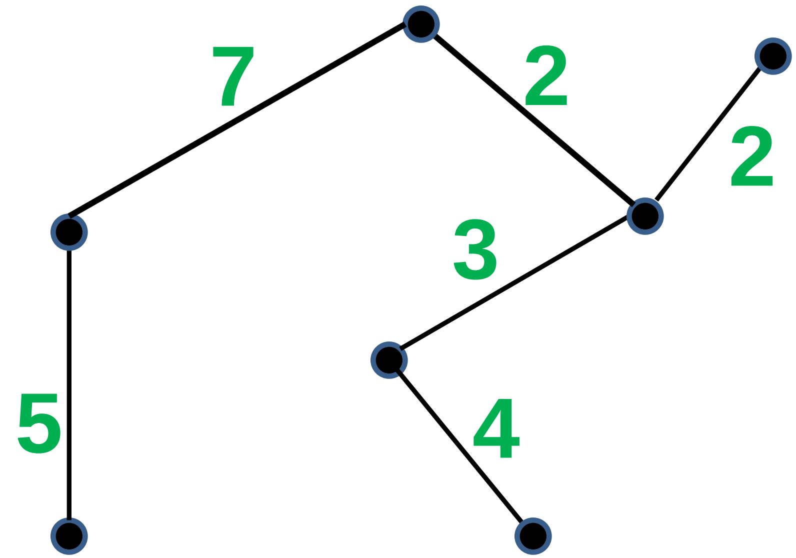
Sort edges in increasing order of their cost

Consider edges in sorted order

If an edge can be added to T without adding a cycle then add it to T

Kruskal's Algorithm





Input: G=(V,E), c_e> 0 for every e in E

T = Ø

Sort edges in increasing order of their cost

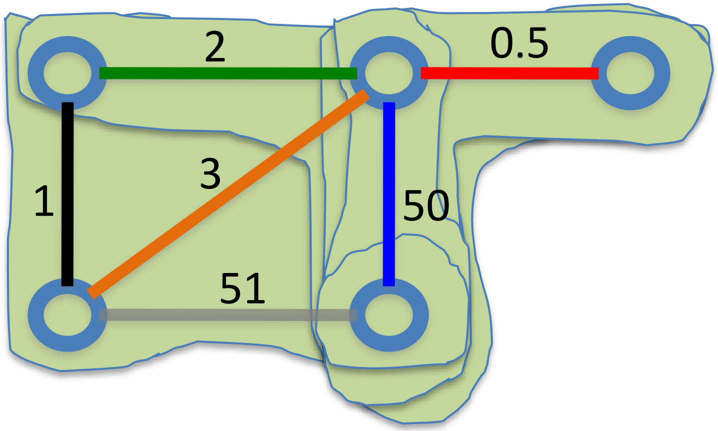
2, 2, 3, 4, 4, 5, 5, 7, 7, 7, 9, 12

Consider edges in sorted order

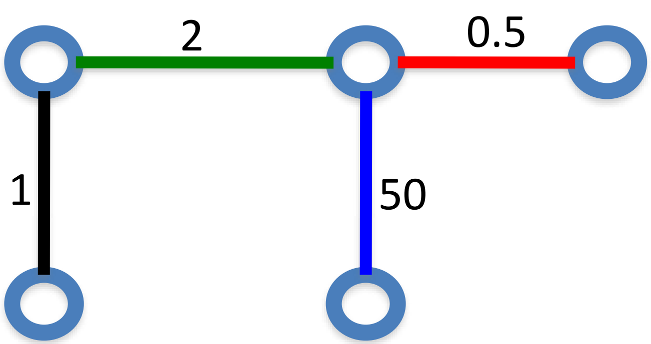
If an edge can be added to T without adding a cycle then add it to T

Prim’s algorithm

Similar to Dijkstra’s algorithm



Robert Prim



Input: G=(V,E), $c_e > 0$ for every e in E

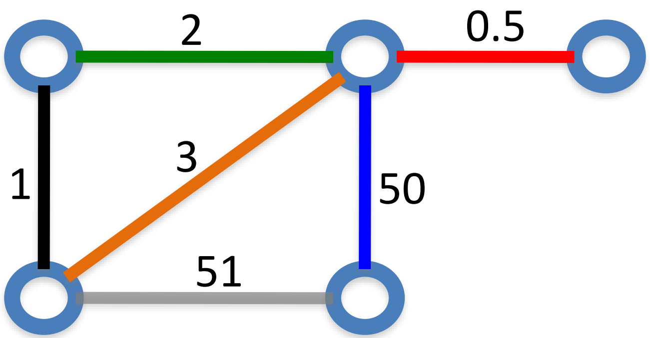
S = {s}, T = Ø

While S is not the same as V

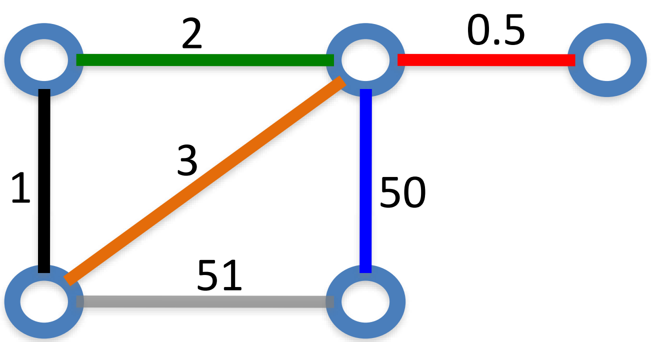
Among edges e= (u,w) with u in S and w not in S, pick one with minimum cost

Add w to S, e to T

Reverse-Delete Algorithm



Input: G=(V,E), c_e> 0 for every e in E



T = E

Sort edges in decreasing order of their cost

Consider edges in sorted order

If an edge can be removed T without disconnecting T then remove it