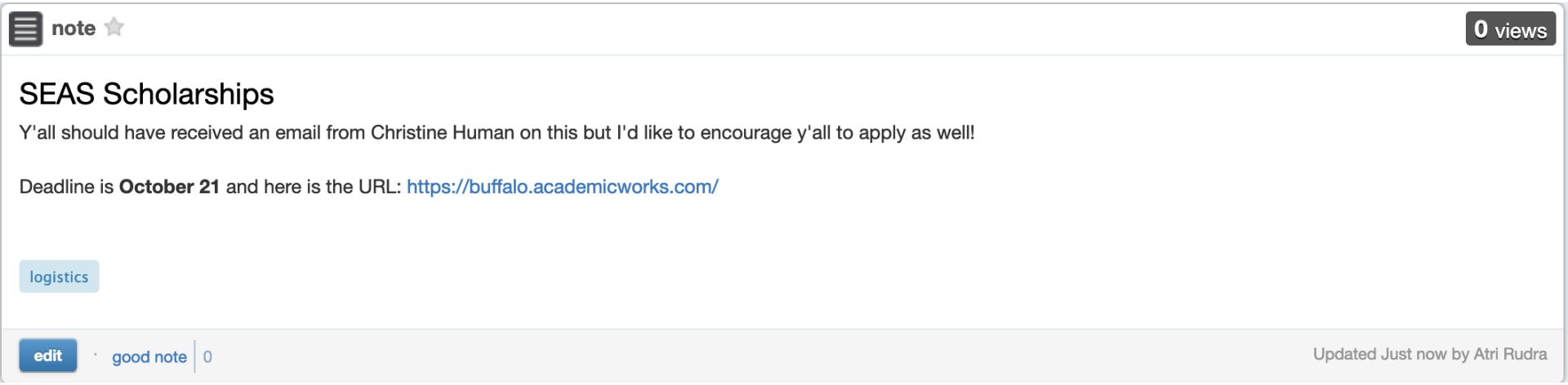
Lecture 20

CSE 331

Oct 11, 2019

SEAS scholarships



HW 5 solutions

At the end of the lecture

Extra OH today

note

stop following 87 views

Extra Friday OH

A gentle reminder that you will be able to pickup solutions to **all** HWs (1-5) during the office hours **this** Friday (all in **Salvador Lounge)**:

- **Elijah** (as usual) will have his OH **3-4:10pm** (note the extra 20 mins beyond the usual 3:50pm stop).
- **Nick** will have a special OH **5-5:50pm**.

We *might* add another OH earlier in the day: I'll update this post in case we are able to do so.
#pin

mid-term office_hours

edit · good note | 0

Updated 22 hours ago by Atri Rudra

Graded HW 4 and Quiz 1

Planning for tonight

Mid-term-I on Monday

1-1:50pm in this place

If you can reference away a Q, do it!

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

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

$T = \emptyset$

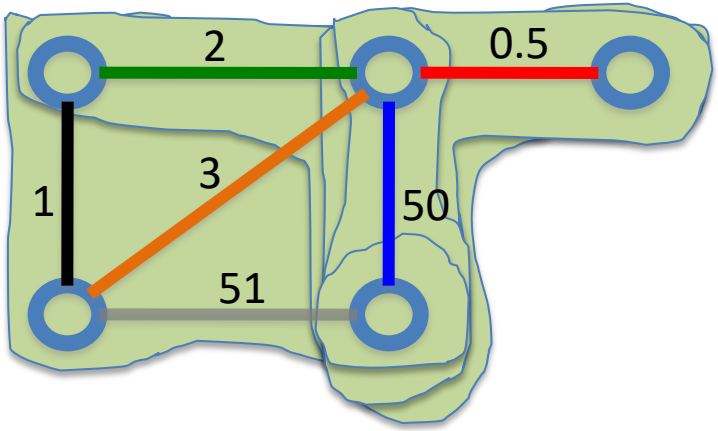
Sort edges in increasing order of their cost

Consider edges in sorted order

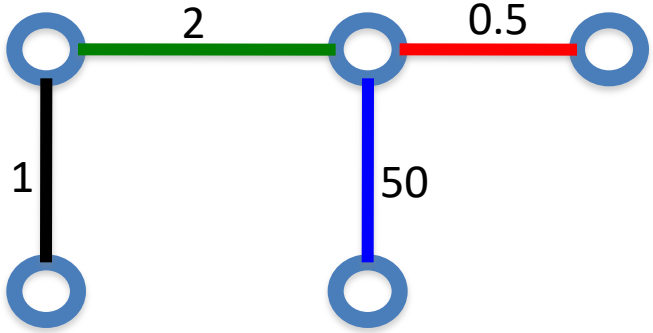
If an edge can be added to T without adding a cycle then add it to T

Prim's algorithm

Similar to Dijkstra's algorithm



Robert Prim



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

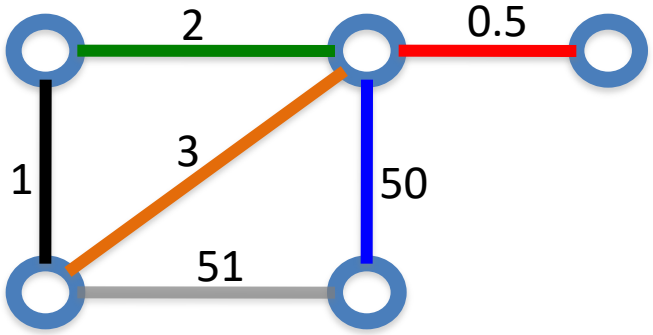
S = {s}, T = Ø

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



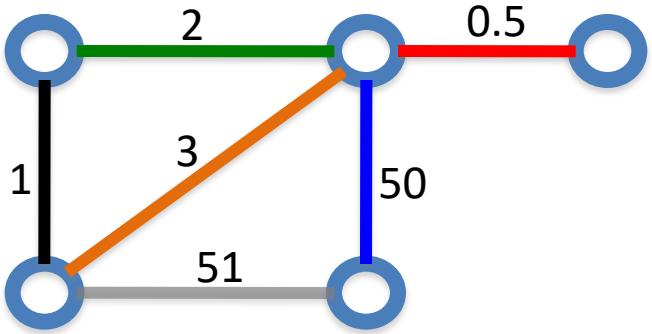
Input: G=(V,E), $c_e > 0$ for every e in E

T = E

Sort edges in decreasing order of their cost

Consider edges in sorted order

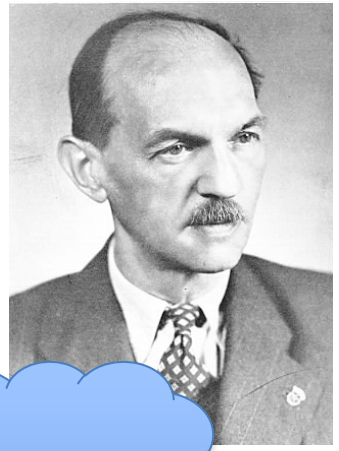
If an edge can be removed T without disconnecting T then remove it



(Old) History of MST algorithms

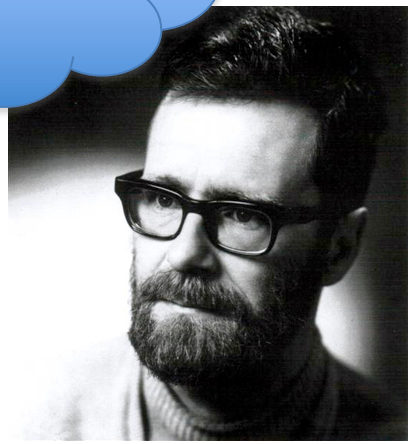
1920: Otakar Borůvka

1930: Vojtěch Jarník



Same algo!

1956: Kruskal



1957: Prim

1959: Dijkstra