MINIMUM SPANNING TREES

David Kauchak
CS62– Fall 2020

1

Quiz!

2

Admin

Assignment 10 due Tuesday

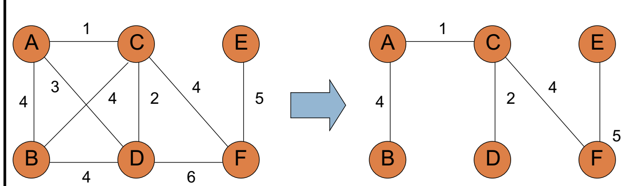
Tuesday's class
- Recap
- Bring questions!

Final exam 12/4, 2-5pm (PST)
- More details on Tuesday

3

Minimum spanning trees (MST)

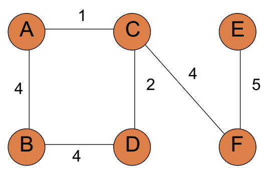
The lowest weight set of edges that connects all vertices of an undirected graph with positive weights



4

MSTs

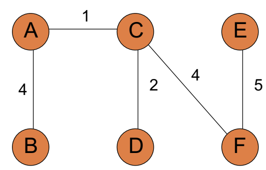
Can an MST have a cycle?



5

MSTs

Can an MST have a cycle?



6

Applications?

Connectivity

- Networks (e.g. communications)
- Circuit design/wiring

hub/spoke models (e.g. flights, transportation)

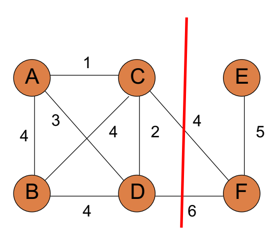
approximate solutions to some "hard" problems

7

Cuts

A cut is a partitioning of the vertices into two sets S and V-S

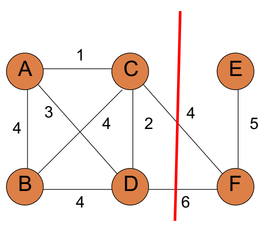
An edge "crosses" the cut if it connects a vertex u∈V and v∈V-S



8

Minimum cut property

Given a partition S, let edge e be the minimum cost edge that **crosses** the partition. *Every* minimum spanning tree contains edge e.

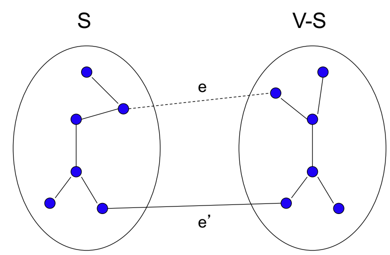


Why?

9

Minimum cut property

Given a partition S, let edge e be the minimum cost edge that **crosses** the partition. *Every* minimum spanning tree contains edge e.

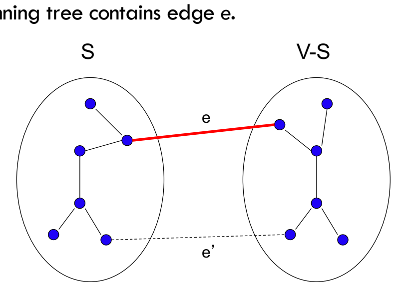


Consider an MST with edge e' that is not the minimum edge

10

Minimum cut property

Given a partition S, let edge e be the minimum cost edge that **crosses** the partition. *Every* minimum spanning tree contains edge e.

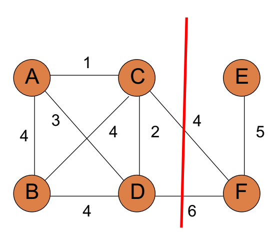


Using e instead of e', still connects the graph, but produces a tree with smaller weights

11

Minimum cut property

If the minimum cost edge that **crosses** the partition is not unique, then *some* minimum spanning tree contains edge e.



12

Kruskal's algorithm

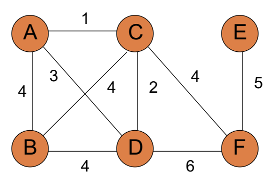
Given a partition S, let edge e be the minimum cost edge that **crosses** the partition. *Every* minimum spanning tree contains edge e.

Kruskals:
- Sort edges by increasing weight
- for each edge (by increasing weight):
  - check if adding edge to MST creates a cycle
  - if not, add edge to MST

13

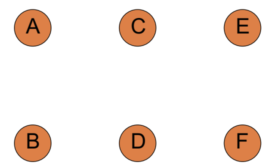
Kruskal's algorithm

Add smallest edge that doesn't create a cycle



G

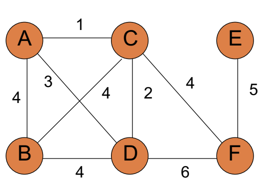
MST



14

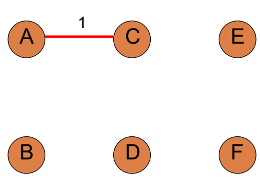
Kruskal's algorithm

Add smallest edge that doesn't create a cycle



G

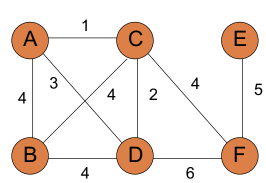
MST



15

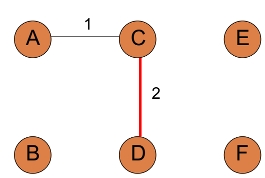
Kruskal's algorithm

Add smallest edge that doesn't create a cycle



G

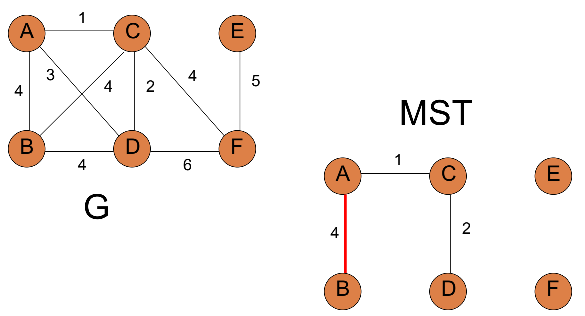
MST



16

Kruskal's algorithm

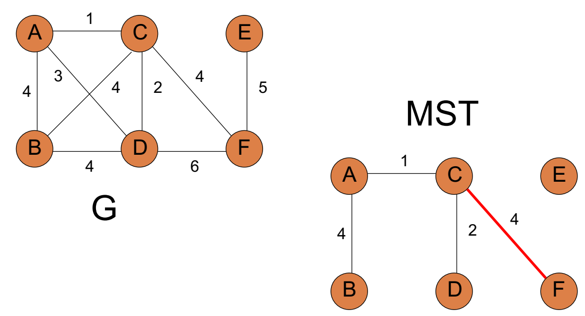
Add smallest edge that doesn't create a cycle



17

Kruskal's algorithm

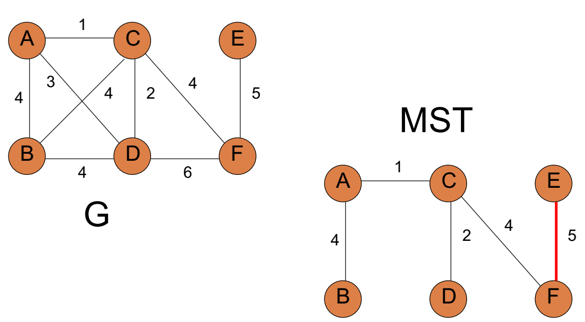
Add smallest edge that doesn't create a cycle



18

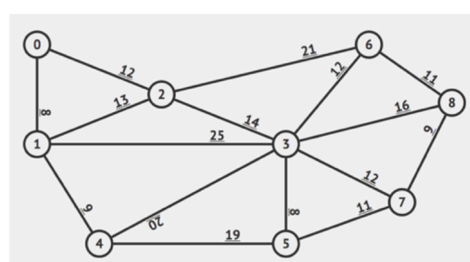
Kruskal's algorithm

Add smallest edge that doesn't create a cycle



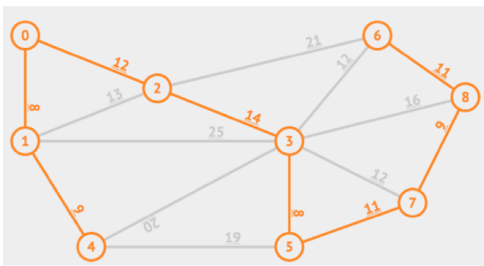
19

Practice



20

Solution



Sum = 8 + 8 + 9 + 9 + 11 + 11 + 12 + 14 = 82

21

Why does Kruskal's work?

Never adds an edge that creates a cycle

Therefore, always adds lowest cost edge to connect two connected components. By min cut property, that edge must be part of the MST

Kruskals:
- Sort edges by increasing weight
- for each edge (by increasing weight):
  - check if adding edge to MST creates a cycle
  - if not, add edge to MST

22

Why does Kruskal's work?

Never adds an edge that creates a cycle

Therefore, always adds lowest cost edge to connect two connected components. By min cut property, that edge must be part of the MST

Kruskals:
- Sort edges by increasing weight
- for each edge (by increasing weight):
  - check if adding edge to MST creates a cycle
  - if not, add edge to MST

Run-time?

23

Why does Kruskal's work?

Never adds an edge that creates a cycle

Therefore, always adds lowest cost edge to connect two connected components. By min cut property, that edge must be part of the MST

Kruskals:
- Sort edges by increasing weight
- for each edge (by increasing weight):
  - check if adding edge to MST creates a cycle
  - if not, add edge to MST

Run-time: O(V+E) to do a DFS/BFS

24

Why does Kruskal's work?

Never adds an edge that creates a cycle

Therefore, always adds lowest cost edge to connect two connected components. By min cut property, that edge must be part of the MST

Kruskals:
- Sort edges by increasing weight
- for each edge (by increasing weight):
  - check if adding edge to MST creates a cycle
  - if not, add edge to MST

Overall run-time?

25

Why does Kruskal's work?

Never adds an edge that creates a cycle

Therefore, always adds lowest cost edge to connect two connected components. By min cut property, that edge must be part of the MST

Kruskals:
- Sort edges by increasing weight
- for each edge (by increasing weight):
  - check if adding edge to MST creates a cycle
  - if not, add edge to MST

Run-time: $O(VE+E^2)$ do this E times

26

Kruskal's details

We can do better!

Uses a data structure called "disjoint set" to efficiently check whether adding an edge creates a cycle

Run-time: O(E log E) (bounded by the sort)

27