Algorithm Design Techniques

- Induction
- Divide and Conquer
- Dynamic Programming
- **Greedy**
- Reduction

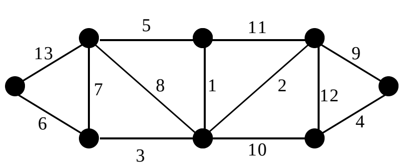
Greedy Paradigm
Get what you can NOW!

But sometimes it's better to look around!

But sometimes it isn't …

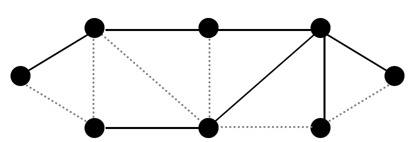
Minimum Spanning Tree

- G=(V,E) is a connected, weighted graph with n vertices and m edges.



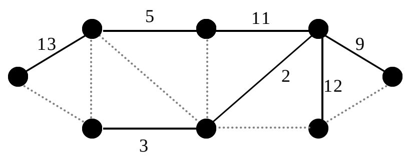
Minimum Spanning Tree-cont.

- A spanning tree of G is a connected, acyclic subgraph with vertex set V.



Minimum Spanning Tree-cont.

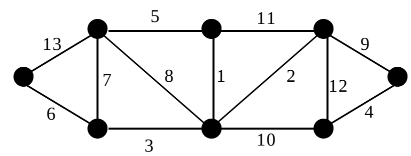
- The weight of spanning tree of G is the sum of the weights of its edges.



CS140 18-7

Minimum Spanning Tree-cont.

- A minimum spanning tree of G is one with smallest possible weight.
- Find an MST of the following graph:



CS140 18-8

Kruskal's (Greedy) Algorithm

Let $e_1, e_2, \dots, e_m$ be the edges of G sorted by increasing weight.

F=V (F is a forest of isolated vertices)

For i=1 to m

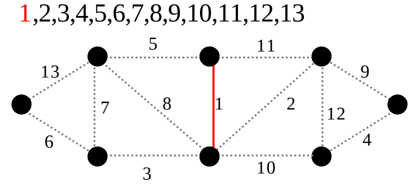
  If F+{e_i} is acyclic then F=F+{e_i}.

Return(F)

CS140 18-9

Kruskal's Algorithm

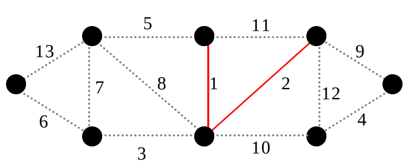
- Order the edge weights. (In this graph the weights are unique.)
- 1,2,3,4,5,6,7,8,9,10,11,12,13



CS140 18-10

Kruskal's Algorithm-cont.

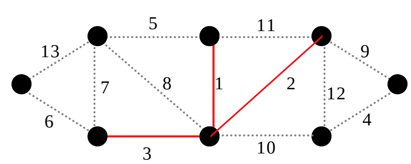
- 1,2,3,4,5,6,7,8,9,10,11,12,13



CS140 18-11

Kruskal's Algorithm-cont.

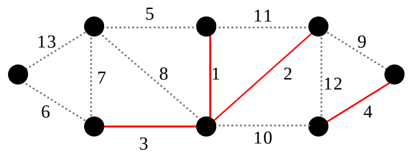
- 1,2,3,4,5,6,7,8,9,10,11,12,13



CS140 18-12

Kruskal's Algorithm-cont.

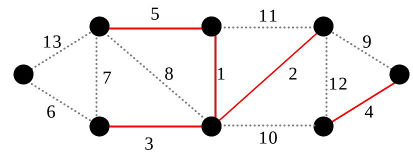
- 1,2,3,4 ,5,6,7,8,9,10,11,12,13



CS140 18-13

Kruskal's Algorithm-cont.

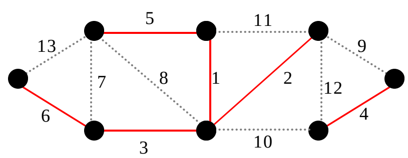
- 1,2,3,4,5,6,7,8,9,10,11,12,13



CS140 18-14

Kruskal's Algorithm-cont.

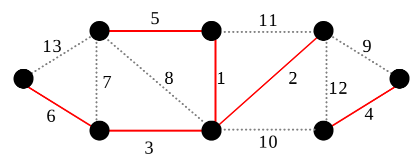
- 1,2,3,4,5,6 ,7,8,9,10,11,12,13



CS140 18-15

Kruskal's Algorithm-cont.

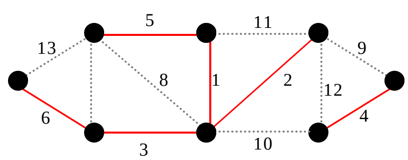
- 1,2,3,4,5,6,**7**,8,9,10,11,12,13
- Can we add the edge with weight 7?



CS140 18-16

Kruskal's Algorithm-cont.

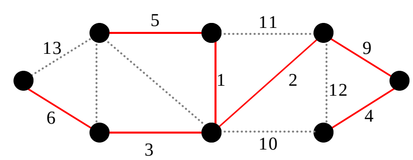
- 1,2,3,4,5,6,7,8,9,10,11,12,13
- Can we add the edge with weight 8?



CS140 18-17

Kruskal's Algorithm-cont.

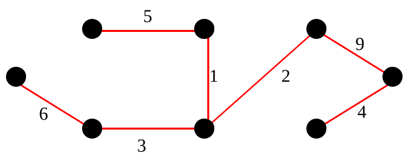
- 1,2,3,4,5,6,7,8,9,10,11,12,13



CS140 18-18

Kruskal’s Algorithm-cont.

MST of G with cost ________



CS140 18-19

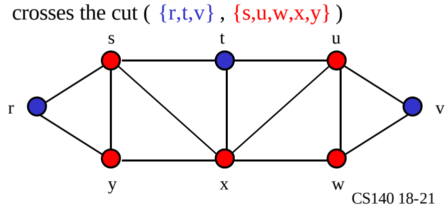
Kruskal’s Algorithm

- Does it work in general?
- Prove it.

CS140 18-20

Cut

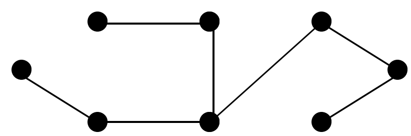
A **cut** is a partition of the vertices of G into two sets (S,S’). An edge e crosses the cut if it has an endpoint in each set of the cut. For example (t,x) crosses the cut ({r,t,v} , {s,u,w,x,y})



CS140 18-21

Tree Facts

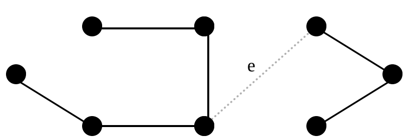
- A tree on n nodes has n -1 edges.



CS140 18-22

Tree Facts

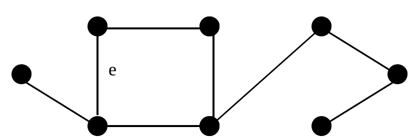
- If e is an edge of T then T-{e} is a forest consisting of two trees.



CS140 18-23

Tree Facts

- If e is an edge of G but not of T then T+{e} contains exactly one cycle.



CS140 18-24

Tree Facts

1. A tree on n nodes has n -1 edges.
2. If e is an edge of T then T-{e} is a forest consisting of two trees.
3. If e is an edge of G but not of T then T+{e} contains exactly one cycle.

CS140 18-25

Kruskal's Algorithm Proof of Correctness

Claim:

At each stage of the algorithm F is a subgraph of some MST of G.

CS140 18-26

Kruskal's Algorithm

Let $e_1, e_2, \dots, e_m$ be the edges of G sorted by increasing weight.

F=V (F is a forest of isolated vertices)

Claim is true here

For i=1 to m

If F+{e_i} is acyclic then F=F+{e_i }.

Return(F)

CS140 18-27

Loop Invariant

Let $e_1, e_2, \dots, e_m$ be the edges of G sorted by increasing weight.

F=V (F is a forest of isolated vertices) Claim is true here

If Claim is true here

For i=1 to m

If F+{e_i} is acyclic then F=F+{e_i}.

Then Claim is true here

Return(F)

CS140 18-28

Kruskal's Algorithm Proof of Correctness

Loop Invariant:

F is a subgraph of some MST of G .

Proof:

Consider the k[th] execution of the loop. Let T be a MST of G containing F. What can happen during the loop?

1. e_k is not added to F
2. e_k is added to F

CS140 18-29

Kruskal's Algorithm Proof of Correctness

Loop Invariant:

F is a subgraph of some MST of G.

Proof:

1. e_k is not added to F

   In this case F does not change so the claim holds when execution of loop concludes

CS140 18-30

Kruskal's Algorithm Proof of Correctness

Loop Invariant:

F is a subgraph of some MST of G.

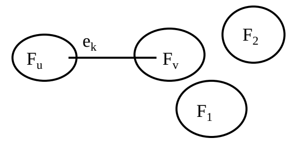
Proof:

2. e_k is added to F
   We know that T is a MST of G and T contains F.
   Need to show there is a MST of G that contains F+{ e_k }
   If e_k is an edge of T we are done. So assume not.

CS140 18-31

What do we know?

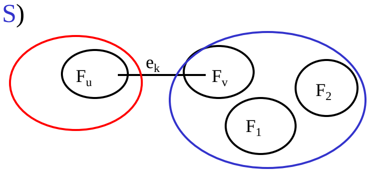
Assume e_k=(u,v). The vertices u and v are in separate connected components. Let S be the vertices of F_u.



CS140 18-32

What do we know?

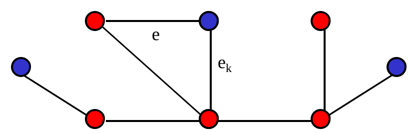
e_k is a minimum weight edge spanning (S,V-S)



CS140 18-33

Using our tree facts

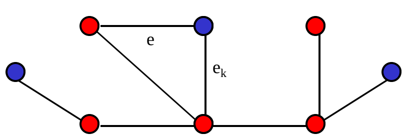
- The graph T+{e_k} contains exactly one cycle.
- This cycle contains e_k and at least one additional edge e that spans (S,V-S).
- T+{e_k}-{e} is an MST of G.



CS140 18-34

Moreover

- T+{e_k}-{e} is an MST of G **that contains the edges of F+{e_k}.**



CS140 18-35