Announcements

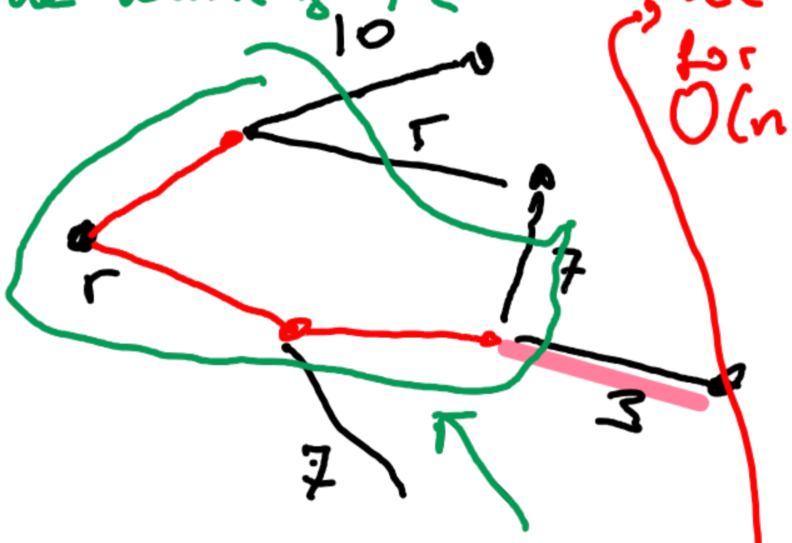
- Hw1 ~~is~~ will be available on Gradescope after class. Due Friday Jan 30
- Check out all course policies on the policy web page: https://www.cs.cornell.edu/courses/cs4820/2026sp/policies/
- Sections start (Monday/Tuesday), attendance mandatory
- Course announcements will be pinned on our Ed page.
- Office hours listed on course web page. Starting now

More in Min cost spanning tree (MST)

Last time: Kruskal's algorithm finds a min-cost spanning tree.

Today: Prim's algorithm finds a min-cost spanning tree. Same proof structure ("exchange argument")

start node r
  while selected edges not tree (on the whole graph)
  add min cost edge
    connecting r's component
    to a new node



see book for $O(n \log m)$

Running time: say $|V| = n$, $|E| = m$

$\le n$ - iterations of while loop $(n-1)$
$O(m)$ to implement an iteration
} $O(n \cdot m)$ (*)

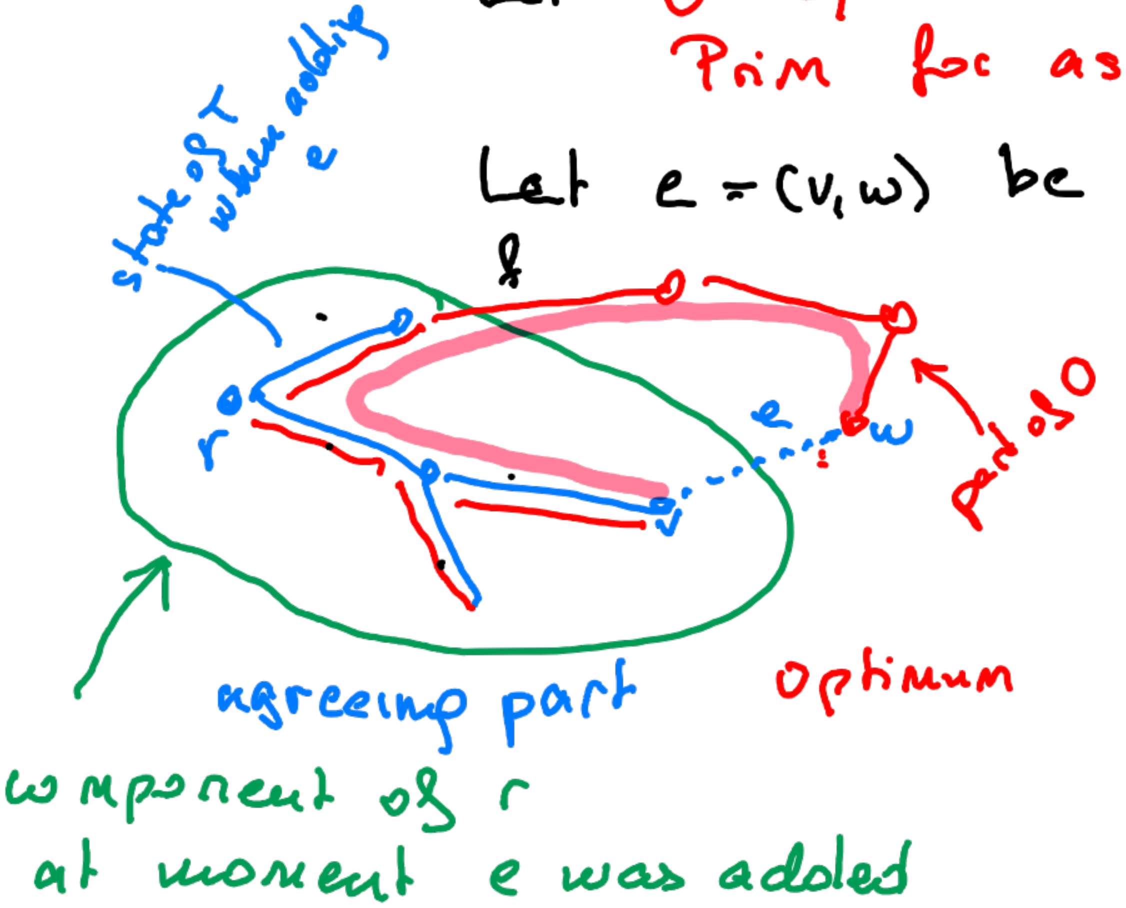
Proving Prim finds the optimal solution

Suppose not true

Let T tree found by Prim

Let O optimal solution, that agrees with Prim for as long as possible

Let e = (v, w) be first edge T not in O (alg made mistake)



Opt connected:
⟹ must have v ~ w path

Goal (exchange)
pick f O − f + e claim contradiction

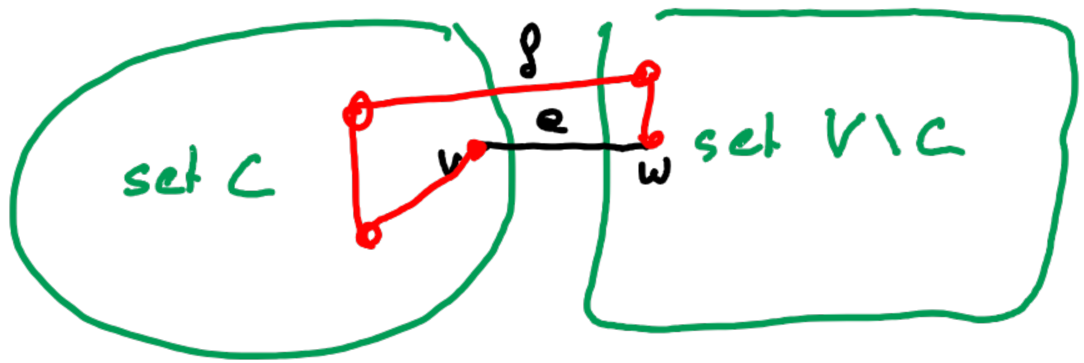
Need
- $c_f \geq c_e$
- O − f + e connected

Pick f that on path on O from v to w
that leaves the component of r when e was added

Claim f has two properties above

1. by choice of e $c_f \geq c_e$
2. O's path from v to w + edge e = cycle
   see last time: deleting any edge from a cycle keeps graph connected

More generally: cut property



consider cheapest edge in cut $e = (v, w)$

Claim: ① there exist a min cost spanning tree containg e

② if all c_e all different then min cost spanning tree must contain e

Proof: suppose not: O optimum $e \notin O$

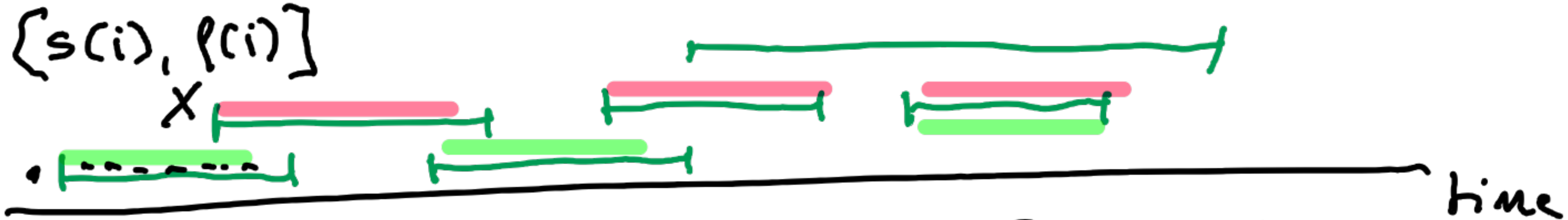
there is v to w path in O

select edge f in path going from C to V\C

$O + e - f$

Greedy II: Interval scheduling.

The problem: request times on a single resource



problem: input n request intervals {s(i), f(i)}
find as many disjoint intervals as possible

Ideas for greedy algorithms:

A. earliest finish time

B. select one with fewest conflict

C. shortest interval length

D. earliest start

E. remove one with most conflicts

Join by Web **PollEv.com/evatardos772**

see next time

A. earliest finish time

B. ✗ select one with fewest conflict

C. ✗ shortest interval length

D. ✗ earliest start

E. ✗ remove one with most conflicts

F. all working