`http://www.cs.unh.edu/~ruml/cs758`

Spanning Trees

Problems

lightest total, lightest max, heaviest, . . .

network connectivity
power, water distribution
wiring, VLSI

number of edges?
cycles?

Basic Approach

starting from $\emptyset$, grow spanning tree by adding edges

Basic Approach

starting from $\emptyset$, grow spanning tree by adding edges

Theorem: take any cut that respects the nascent tree. A lightest edge crossing the cut can be added to the tree.

Basic Approach

starting from $\emptyset$, grow spanning tree by adding edges

Theorem: take any cut that respects the nascent tree. A lightest edge crossing the cut can be added to the tree.

Proof: if a MST T includes our edge, fine. Otherwise, consider an edge in T that crosses cut. Replace it with ours. Still a spanning tree. Cost can't go up, so still minimum.

Kruskal's Algorithm

The Algorithm

connect separate components until spanned

The Algorithm

connect separate components until spanned

1. $T \leftarrow \emptyset$
2. for each vertex v, $\textsc{Make-Set}(v)$
3. for each edge (u, v) in nondecreasing order of weight
4.   if $\textsc{Find-Set}(u) \neq \textsc{Find-Set}(v)$
5.   add edge to T
6. $\textsc{Union}(u, v)$
7. return T

correctness?
running time?

Break

- asst 8
- asst 9

Prim's Algorithm

The Algorithm

grow tree until connected

The Algorithm

Spanning Trees
Kruskal's Algorithm
Prim's Algorithm
- Algorithm
- Random ?s
- EOLQs

grow tree until connected

for each vertex v, $v.c \leftarrow \infty$ and $v.\pi \leftarrow$ nil
$1.c \leftarrow 0$
$Q \leftarrow$ heap of all vertices
while Q is not empty
    $u \leftarrow$ remove vertex with minimum c
    for each neighbor v of u
        if v is in Q and $w(u, v) < v.c$
            $v.c \leftarrow w(u, v)$
            $v.\pi \leftarrow u$
return $\{(u, u.\pi) : u \in V - \{1\}\}$

correctness? what is the invariant?
running time?

Random Questions

Let G be an undirected connected graph in which all edge weights are distinct. Which of these are true?

1. Every MST of G contains the edge of minimum weight.
2. No MST contains the edge of maximum weight.
3. If the edge of maximum weight were in an MST, then removing it would disconnect G.
4. G has a unique MST.

EOLQs

For example:

- What's still confusing?
- What question didn't you get to ask today?
- What would you like to hear more about?

Please write down your most pressing question about algorithms and put it in the box on your way out.
Thanks!