Set Cover:

Input: A collection X of n elements

& m subsets $\{Y_1, Y_2, \ldots, Y_m\}$ so $(Y_i \subseteq X)$. Assume

$$\bigcup_{i=1}^{m} Y_i = X.$$

Goal: Pick as few subsets

$\{Y_{i_1}, Y_{i_2}, \ldots, Y_{i_r}\}$ as possible

so $$\bigcup_{j=1}^{r} Y_{i_j} = X.$$

Claim: Set Cover is NP-hard.

(from Min Vertex Cover)

Given undirected $G=(V,E)$.

Want a smallest collection of vertices touching all edges.

(covering)

Create an instance of Set Cover...

$X := E$.

For each vertex $v \in V$, make a subset Y_v = incident edges of v.

Solve Set Cover & return vertices corresponding to the best collection of subsets.

Takes $O(V+E)$ time.

Min Vertex Cover is just a special case of set cover; I just detailed how.

-or-

Given a vertex cover, it cover all edges, so the subsets of incident edges include all edges, so these subsets are a set cover.

Given a set cover, the subsets are the edges incident to corr. vertices, so these vertices cover all edges & are a vertex cover.

Decision: Given rest of input + an integer k, is there a set cover with k subsets.

$\in NP$: Certificate is the k subsets. Can verify in poly time that they cover X.

$\Rightarrow$ decision version is NP-complete.

Spring 2019 6b:

Given a complete undirected graph K_n over n vertices with non-negative weights on edges, find a min-weight cycle that includes every vertex.

Prove it is NP-hard.

Use Ham. cycle in undirected graphs.

Given undirected $G = (V, E)$.

Create $K_{|V|}$.

Weigh each edge uv as 0 if $uv \in E$.

$+1$ otherwise.

Return whether min cost cycle has weight 0.

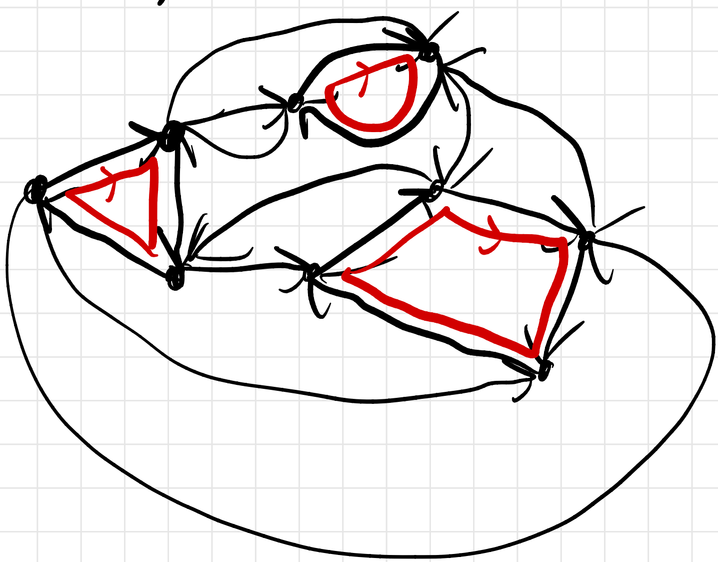
Time: $O(V^2)$ need to write weights for all $O(V^2)$ edges

If G has a Ham. cycle C, all its edges have weight 0 in $K_{|V|}$, so C is a 0 weight cycle that includes every vertex.

If $K_{|V|}$ has a 0 weight cycle C including all vertices, all edges are from G. So C is a Ham. cycle in G.

Practice QE 4.2:

A cycle cover of directed $G=(V,E)$ is a set of vertex-disjoint cycles that cover (include) every vertex in G. (Each cycle has ≥ 2 vertices.)



Goal: Find a cycle cover or argue none exists.

Obs: Every vertex needs exactly one outgoing edge.

Obs: Every vertex needs exactly one incoming edge.

in the cycle cover

A subset of edges is a cycle cover iff it meets those conditions.

- equivalent to picking one successor vertex for each vertex + one predecessor.

- or pair up vertices into predecessor-successor pairs

So, make bipartite graph

$G' = (V', E')$.

$V' = V \times \{p, s\}$

$E' = \{(u,p)\,(v,s) \ : u \to v \in E\}$

Compute a max bipartite matching M in G'.

If it includes every vertex in G', cycle cover is $\{u \to v : (u,p)(v,s) \in M\}$

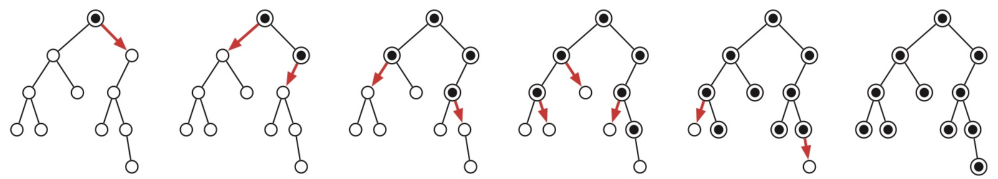
o.w., report no cycle cover.

time: $|V'| = 2|V|$

$|E'| = |E|$

Takes $O(V'E') = O(VE)$ to build G' & compute matching.

2019 Q4: Given n-node binary tree. Root knows a message. In a single round, any node with message can forward it to at most one child.



What min # rounds to broadcast to all nodes.

Broadcast Time (v): min # rounds for v to broadcast a known message to all nodes in its subtree

BroadCast Time(v) =

0 if v is a leaf

1 + BroadCast Time(c) if v has one child c

2 + BroadCast Time(l) if Broadcast Time(l) = Broadcast Time(r) where v has children l or r.

1 + Broadcast Time(s) if Broadcast Time(s) $\neq$ Broadcast Time(f) where v has children s & f.

Return Broadcast Time(r) where r is the root of input tree.

Compute all n solutions in postorder.

$O(1)$ time per subproblem,

so $O(n)$ time total.

first recipient

really takes $\max\{1 + \text{BroadcastTime}(x), 2 + \text{BroadcastTime}(y)\}$

if you have children x & y.

if $\text{BroadcastTime}(x) \geq \text{BroadcastTime}(y)$,

this equals $1 + \text{BroadcastTime}(x)$.

if $BT(x) \lneq BT(y)$, equals $2 + \text{BroadcastTime}(y)$