Set Cover:
Input: A collection \( \mathbf{X} \) 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
\[ \{Y_{\tilde{1}}, Y_{\tilde{2}}, \ldots, Y_{\tilde{r}}\} \] as possible
so
\[ \bigcup_{j=1}^{r} Y_j = X. \]
Claim: Set Cover is \( \mathsf{NP} \)-hard.
(from Min Vertex Cover)
Given undirected $G = (V, E)$. Want a smallest collection of vertices touching all edges.

Create an instance of Set Cover...

$X := E$.

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

Solve Set Cover & return vertices corresponding to the best collection of subsets. Takes $O(\text{V+\text{E}})$ 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 and an integer $k$, is there a set cover with $k$ subsets?

$\in \text{NP}$: Certificate is the $k$ subsets. Can verify in poly time that they cover $\times$.

$\implies$ decision version is NP-complete.
Spring 2019 66:

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 $uv \notin E$, $+1$ otherwise.
Return whether min cost cycle has weight 0.

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

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

If $K_{n,n}$ 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 $\geq 2$ 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 if 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

\[ G' = (V', E') \]

\[ V' = V \times \{ p, s \} \]

\[ E' = \{(u, p) (v, s) : u \Rightarrow v \in E \} \]

Compute a max bipartite matching \( \mathcal{M} \) in \( G' \).

If \( \mathcal{M} \) includes every vertex in \( G' \), cycle cover is \( \{ u \Rightarrow v : (u, p), (v, s) \in \mathcal{M} \} \).

O.w., report no cycle cover.
time: $|V'| = 2|V|$

$|E'| = |E|$

Takes $O(|V'|E'|) = O(|V|E)$ to build $G'$ and compute matching.
2019 Q4: Given a n-node binary tree. Root knows a message. In a single round, any node with message can forward it to at most one child.

With 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 and r.

1 + Broadcast Time (s) if Broadcast Time (s) ≥ Broadcast Time (l) where v has children s and f.

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

Compute all n solutions in post order.
$O(1)$ time per subproblem, so $O(n)$ time total. The first recipient really takes $\max \{ 1 + \text{Broadcast Time} (x), 2 + \text{Broadcast Time} (y)\}$ if you have children $x + y$.

If $\text{Broadcast Time} (x) \geq \text{Broadcast Time} (y)$, this equals $1 + \text{Broadcast Time} (x)$.

If $\text{BT}(x) \neq \text{BT}(y)$, equals $2 + \text{Broadcast Time} (y)$.