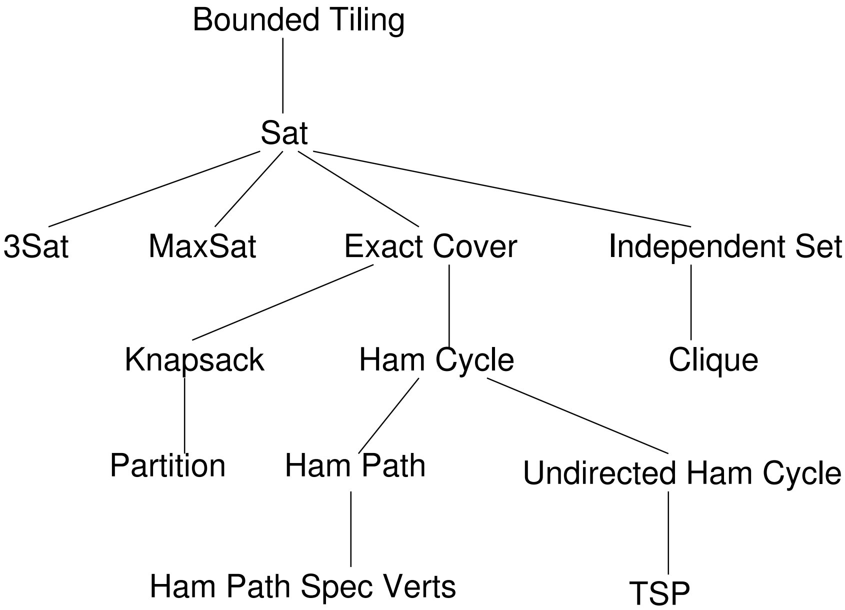
A. Prove $L \in \mathcal{NP}$

1. Describe a certificate.
2. Demonstrate it can be used to check a string/solution in polynomial time.
3. Demonstrate that the certificate iteslf is succinct (polynomial in size)

usually easy for our problems—ok to do briefly/informally

B. Prove L is $\mathcal{NP}$-hard

1. Choose a known $\mathcal{NP}$-complete problem L_2.
2. Describe a reduction τ from L_2 to L.
3. Demonstrate τ can be computed in polynomial time. (Also usually easy.)
4. Demonstrate that $x \in L_2$ iff $\tau(x) \in L$


Bounded Tiling
Sat
3Sat
MaxSat
Exact Cover
Independent Set
Knapsack
Ham Cycle
Clique
Partition
Ham Path
Undirected Ham Cycle
Ham Path Spec Verts
TSP

Reducing Sat to Exact Cover:
Suppose $\{c_1, c_2, \ldots c_\ell\}$ is an instance of Sat.
Define the following instance of Exact Cover:

$$\begin{array}{rcll} & & \{x_i\} & \text{for each variable } i \\ \mathcal{U} = & \cup & \{c_j\} & \text{for each clause } j \\ & \cup & \{p_{jk}\} & \text{for each position } k \text{ in clause } j \end{array}$$

$$\mathcal{F} = \begin{array}{llll} \{ & \forall j,k & \{p_{jk}\} & \\ & \forall i & T_{i\top} = \{x_i\} \cup \{p_{jk} \mid \lambda_{jk} = \sim x_i\} & \\ & \forall i & T_{i\bot} = \{x_i\} \cup \{p_{jk} \mid \lambda_{jk} = x_i\} & \\ & \forall j,k & \{c_J p_{jk}\} & \} \end{array}$$

Proof that HamiltonPath is $\mathcal{NP}$-Complete

Proof. *[HamiltonPath is $\mathcal{NP}$.] Suppose $G = (V, E)$ is a graph, an instance of the HamiltonPath. Let $p = \langle u_1, u_2, \ldots u_n \rangle$ be a a sequence of vertices from V, a proposed Hamilton path in G. With any reasonable representation of G, one can check that each vertex in V appears uniquely in p, and that for any pair of vertices u_i, u_{i+1} as they appear in p, the edge (u_i, u_{i+1}) is in E. Moreover, the path p is smaller than the representation of G, so it is succinct.*

[HamiltonPath is $\mathcal{NP}$-hard.] Next, suppose $G = (E, V)$ is an instance of HamiltonCycle. Let $v_1 \in V$ be an arbitrary vertex. Let $G' = (V', E')$ be a new graph such that v_1 is removed and four new vertices are added, that is, $V' = V - \{v_1\} \cup \{v_a, v_b, v_c, v_d\}$; and every edge that is incident on v_1 is replaced with two analogous edges incident on v_b and v_c, and and edges (v_a, v_b) and (v_c, v_d) are added, that is

$$
\begin{aligned}
E' \quad = \quad & (E - \{(v_1, v_x) \mid (v_1, v_x) \in E\}) \\
& \cup \{(v_b, v_x), (v_c, v_x) \mid (v_1, v_x) \in E\} \\
& \cup \{(v_a, v_b), (v_c, v_d)\}
\end{aligned}
$$

This reduction reduction is accomplished by one pass over the edges, which is polynomially computable.

Now, suppose G has a Hamilton cycle, call it $(v_1, v_2, \dots v_{|V|-1}, v_1)$. (As a cycle, it has an arbitrary starting/ending point, so we are free to choose v_1 as the starting point when naming the cycle.) Then G' has a Hamiltonian path $(v_a, v_b, v_2, \dots, v_{|V|-1}, v_c, v_d)$.

Conversely, suppose G' has a Hamiltonian path. Based on how we constructed G' (for example, the only edge going out of v_a is (v_a, v_b), and the only edge going into v_d is (v_c, v_d)), that path must be in the form $(v_a, v_b, v_2, \dots, v_{|V|-1}, v_c, v_d)$. Then G has a Hamiltonian cycle $(v_1, v_2, \dots v_{|V|-1}, v_1)$.

Therefore HAMILTON PATH *is $\mathcal{NP}$-complete.* $\square$

Proof that LONGEST CYCLE is $\mathcal{NP}$-Complete

Proof. *[LONGEST CYCLE is $\mathcal{NP}$.] Suppose $(G = (V, E), K)$ is an instance of LONGEST CYCLE and p is a path that is a proposed cycle of length K. An algorithm to check that p is consistent with E, has no repeated vertices, and has length at least K, is polynomial with any reasonable representation of G. Moreover, since p is no larger than the representation of G, it is succinct.*
[LONGEST CYCLE is $\mathcal{NP}$-hard.] Suppose $(G = (V, E))$ is an instance of HAMILTON CYCLE. Then make an instance of LONGEST CYCLE by letting $K = |V|$, which obviously can be done in polynomial time.
Since $K = |V|$, any cycle of length (at least) K must be a Hamilton cycle, and any Hamilton cycle must have length K.
Therefore LONGEST CYCLE is $\mathcal{NP}$-complete. $\square$

Proof that Subgraph Isomorphism is $\mathcal{NP}$-Complete

Proof. *[Subgraph Isomorphism is $\mathcal{NP}$.] Suppose $(G_1 = (V_1, E_1), G_2 = (V_2, E_2))$ is an instance of Subgraph Isomorphism and f is a function $V_1 \to V_2$ (expressed as a list of pairs where $(v_{1,a}, v_{2,b})$ indicates $v_{1,a} \in V_1$, $v_{2,b} \in V_2$, and $f(v_{1,a}) = v_{2,b}$) proposed as an isomorphism. An algorithm to check that f is a one-to-one function and that for all $(v_{1,a}, v_{1,b}) \in E_1$, $(f(v_{1,a}), f(v_{1,b})) \in E_2$, is polynomial with any reasonable representation of G. Moreover, since $|f| = O(V_1)$, it is succinct.*
[Subgraph Isomorphism is $\mathcal{NP}$-hard.] Suppose $(H = (W, F))$ is an instance of Hamilton Cycle. Then construct a graph $G = (V, E)$ that such that $|V| = |W|$ and $E = \{(w_1, w_2), (w_2, w_3), \ldots (w_{|V|}, w_1)\}$ An algorithm to construct this graph takes $O(V)$ time.
Note that E has only those edges that make a Hamiltonian cycle. Thus G is isomorphic to a subgraph of H iff H has a Hamiltonian cycle.
Therefore Subgraph Isomorphism is $\mathcal{NP}$-complete. $\square$

Reduction from UHC to TSP (LP pg 324).

Differences between UHC and TSP:

- The graph in TSP is *weighted* (interpreted as distances)
- The graph in TSP is *complete*
- A TSP problem has a *budget*

Suppose we have an instance of UHC, an undirected graph $G = (V, E)$. Construct a graph with the same vertices but complete in its edges and with distances

$$d_{i,j} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } (v_i, v_j) \in E \\ 2 & \text{otherwise} \end{cases}$$

Set the budget to $|V|$.

Reduction from EXACT COVER to KNAPSACK (LP pg 325).

Given an instance of EXACT COVER $(\mathcal{U}, \mathcal{F} \subseteq \mathscr{P}(\mathcal{U}))$, construct an instance of KNAPSACK (S, K):

- $S = \{1, 2, \ldots |\mathcal{U}|\}$
- $K = 2^{|\mathcal{U}|} - 1 = \sum_{i=0}^{|\mathcal{U}|-1}$

Interpret each set in $\mathcal{P}(S)$ as a bit vector.

Problem: Consider $S = \{1, 2, 3, 4\}$ and proposed cover $\{\{1, 3\}, \{1, 4\}, \{1\}\}$.

INDEPENDENT SET problem: Given a graph, is there a set of vertices of size k with none adjacent to each other?

Reduction from 3SAT to INDEPENDENT SET (LP pg 326–327.)

Suppose we have an instance of 3SAT, $F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$. WOLOG, suppose each clause has exactly three literals. Construct an instance of INDEPENDENT SET, (G, K) where $K = m$ and $G = (V, E)$ such that

- There is a vertex in V for each literal occurrence (or clause position) $c_{i,j}$.
- $(c_{i,j}, c_{x,y}) \in E$ if either
  - $i = x$ (they are positions in the same clause; this makes a triangle of vertices), or
  - the literals $c_{i,j}$ and $c_{x,y}$ are negations of each other.

Suppose an independent set of size K exists in G. It must include exactly one vertex in each triangle. Make a truth assignment that makes each literal in the set true.
Suppose a satisfying truth assignment exists. Then for each triangle, pick one vertex corresponding to a true literal.