A language A is NP-complete:

a) A is in NP

b) For every L in NP, L <=P A

( b) For some NP-complete B, B <=P A

NP-complete: SATISFIABILITY, 3-SAT, CLIQUE, VERTEX COVER, SUBSET SUM, HAMILTONIAN CYCLE, SUBGRAPH ISOMORPHISM

Theorem: Subset Sum is NP-complete.

Subset Sum = a set of integers and a target integer T. We want a subset of the integers to sum to exactly T.

Subset Sum is in NP. Guess a subset of the numbers. Add them (linear in the number of digits and number of values in the subset). Check that the sum is equal to T.

Show Vertex Cover <= Subset Sum.

We are given … (we are trying to solve a Vertex Cover problem using a Subset Sum solver)

We are given a graph G and a number k. Can we cover the edges of G using at most k vertices? We turn this into a subset sum problem.

T: k2222222….2 (where the number of 2’s is equal to the number of edges of G.

Each number in the set of numbers, one number for each vertex, and one number for each edge.

Vertex numbers:

e1 e2 e3 e4 e5 … em

v1 1 1 0 0 1 1 0 vertex v1 connects to edges e1, e4, e5

v2 1 0 1 1 0 0 0 vertex v2 connects to edges e2 e3

v3 1 0 0 1 0 0 0 vertex v3 connects to only edge e3

Edge numbers

e1 0 1 0 0 0 0 0 edge number as a 1 only in its edge

e2 0 0 1 0 0 0 0

em 0 0 0 0 0 0 1

Claim, there is a set of number that sums to T if and only if we can cover every edge of G with k vertices.

We need k numbers of the vertex numbers to get that first digit to be k. If we sum these k numbers, we get : kd1d2d3...dm where each d is 1 or 2. To get them to have value 2, for each j where dj is 1, we add the edge number for ej.

(Other way), if we get all 2’s, we could not have done that with only the edge numbers, so we have to cover the edges with the vertices chosen from the vertex numbers.

Theorem: Hamiltonian Cycle is NP-complete.

Hamiltonian Cycle = {<G> | G has a simple cycle that hits every vertex.}

Proof: 1) Hamitonian Cycle is in NP. Guess an ordering of the n vertices, check in linear time if there is an edge from each vertex to the next, and from the last to the first.

2) 3-SAT <=P Ham. Cycle

Given a set of m clauses, each with 3 literals, n total variables.

We need to create a graph such that the graph has a Ham. Cycle if and only if the 3-SAT instance has an assignment that satisfies each clause.

G has a hamiltonian cycle if and only if we can assign T/F to the variables to satisfy every clause.

Theorem: Subgraph isomorphism is NP-complete

Proof: a) Subgraph Isomorphism is in NP:

(SI = {<G,H>| H is isomorphic to a subgraph of G}

Guess (can’t just guess a subgraph of G because you can’t verify that H is isomorphic to that subgraph in polynomial time – know one yet knows how to do this). Instead Guess the isomorphism function f: H→G. Now we can check every pair x,y in H and see that if x,y is not an edge in H, then f(x),f(y) is not an edge in G. And if x,y is an edge of H, then f(x), f(y) is an edge of G.

Ham. Cycle <= Subgraph Isomorphism.

Given a graph G, does G have a cycle? Can’t just guess H to be a cycle of n vertices and ask if H is isomorphic to a subgraph of G.

The trick is to create a new G that subdivides every edge of G. (call this G’). Make H be a cycle of 2n vertices. (G’ now has n + m vertices).

H is a subgraph of G’ if and only if G has a Ham Cycle. If G has a Ham Cycle then, when we subdivide the edges of G, the Ham Cycle becomes a cycle of 2n edges. H is isomorphic to a subgraph of G’. The isomorphism function f maps each vertex of H to a vertex of G’ such there there is an edge between consecutive vertices. Since all edges of G were subdivided to produce G’, there is no direct edge between f(x) and f(y) if x and y are not consecutive edges on H. Suppose H is isomorphic to a subgraph of G’. Then there is a cycle of 2n vertices in G’. Every other vertex is one introduced by the subdivision. Removing them gives a cycle of n vertices in G.

(Alternative proof: Prove HC is in NP.

a) Guess both a subset of |E(H)| edges of G and a isomorphism function f between H and G. Check that the isomorphism function holds.

b) Now that we defined subgraph isomorphism differently, the reduction from Ham. Cycle is to take G and create H = a cycle of n vertices. We will ask if H is isomorphic to a subgraph of G by guessing the n edges of G to keep and checking that what is left is isomorphic to H.

)