**Class Notes April 17th**

Computational Complexity

*P vs NP*

* *Can brute force algorithms always be significantly sped up?*

To answer this we can only consider decision problems (Optimization problems are composed of decision problems, so by extension our consideration applies to most types of problems)

*P = Polynomial time*

*A decision problem L is in P if there is an algorithm solving L in time O(n^c) for some c, n = bit length of the input.*

Why P?

- Model Independence

- Closed under composition

…

-but really, because of historical use

Edmonds, “Paths, Trees, and Flowers” - A matching in graph G is a subset of edges E\_m <= E(G) with the property that, within that subset, each vertex v touches at most one edge in E\_m.

*NP = Nondeterministic Polynomial Time*

*Formalization of brute force.*

*A decision problem L is in NP if there is a polytime algorithm V(“verifier”)*

*L(x) = yes iff witness w s.t. V(x,w) = yes*

*|w| <= polynomial p, the bit length of x*

Questions:

What makes a problem in NP?