CS 223 Lec 26

Computability Theory

study what types of problems can / cannot be solved by computers

note: unlimited resources (memory) assumed.

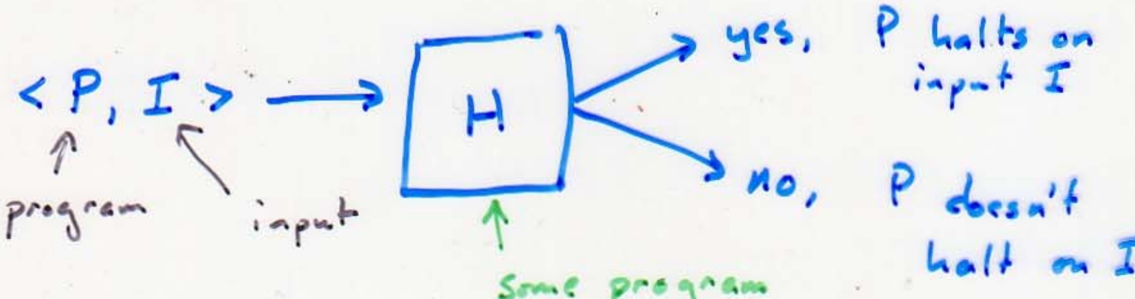
Halting Problem

Given a program P and input I determine if P halts on input I?

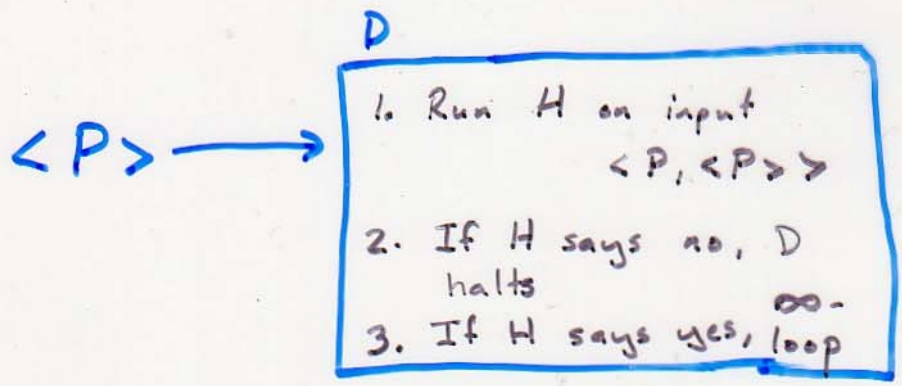
Result: this is an uncomputable problem

Sketch of Proof

By contradiction. Assume that the halting problem is computable. Then $\exists$ another program H



Construct another program called D that uses H as a subroutine.



<D> is a correct (well-specified) program

What happens when we run D on input $\langle D \rangle$?

<u>Two possibilities</u>

1) Suppose $D(\langle D \rangle)$ halts

$\Rightarrow H(\langle D, \langle D \rangle \rangle) \rightarrow$ "no"

$\Rightarrow$ D on input $\langle D \rangle$ doesn't halt.

$\times$

2) Suppose $D(\langle D \rangle)$ loops forever

$\Rightarrow H(\langle D, \langle D \rangle \rangle) \rightarrow$ "yes"

$\Rightarrow$ D on input $\langle D \rangle$ halts

$\times$

$\Rightarrow$ contradiction in both cases

$\Rightarrow$ H cannot exist

$\Rightarrow$ Halting Problem is <u>not</u> computable.

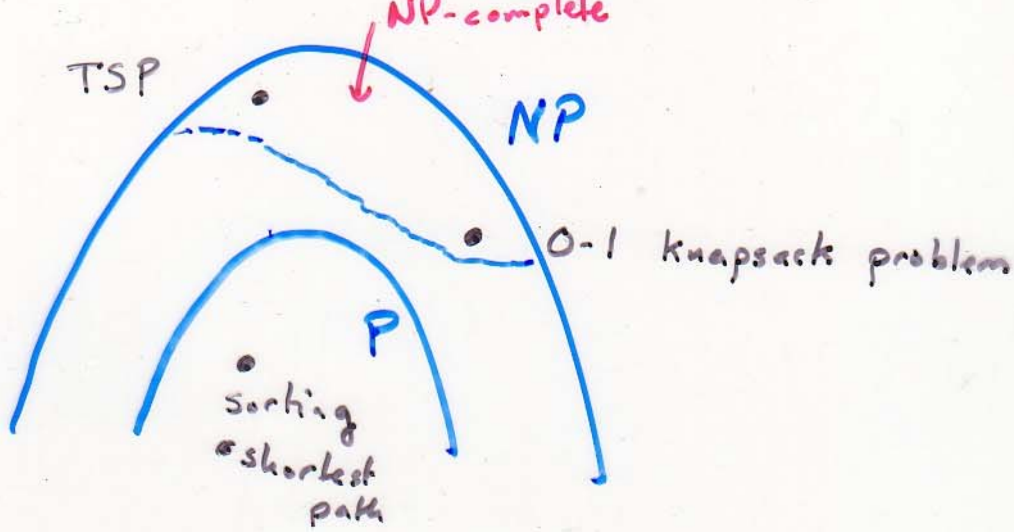
Complexity Theory

studys the inherent complexity of solving computable problems.

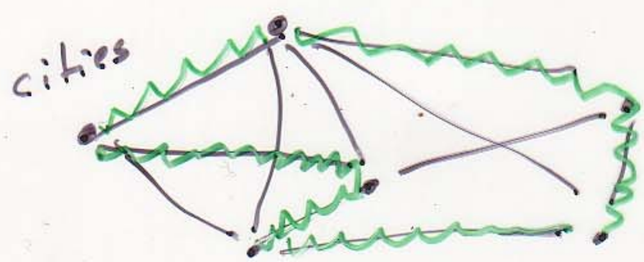
"P vs. NP" question

P = class of problems solvable in polynomial time
(almost all practical algorithms here)

NP = class of problems solvable on a non-deterministic computer in polynomial time.



TSP = travelling salesman problem

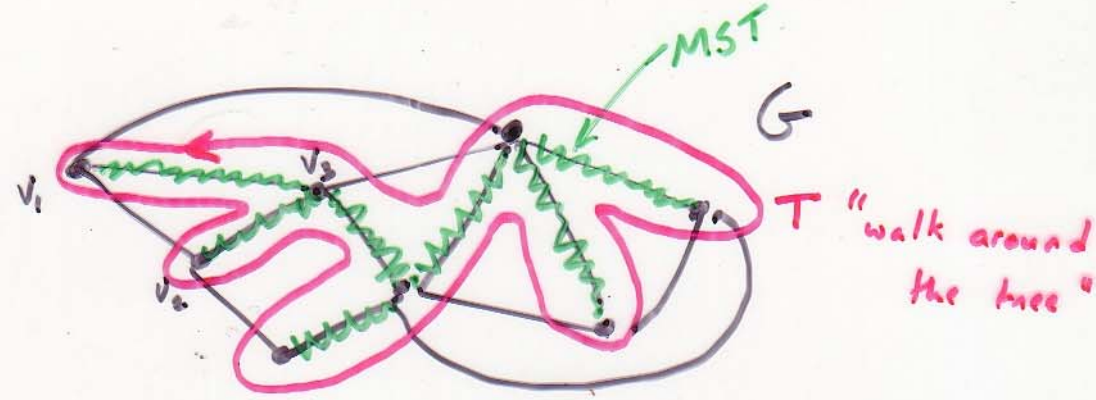


Find the least cost "tour" that visits all cities exactly once.

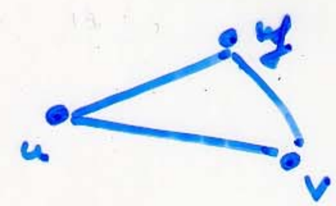
Approximation Algorithms

try to find an algorithm that produces a soln that is within X % of optimal.

An approximation algorithm for TSP



Assume G satisfies the triangle inequality..

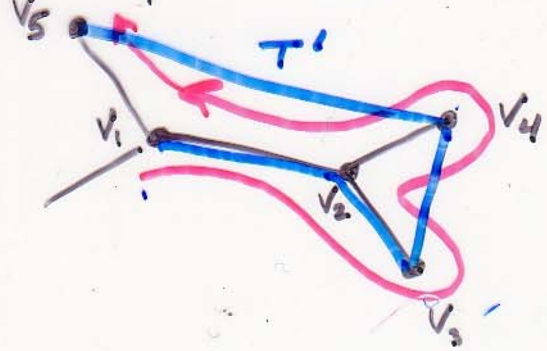


$$w(u,v) \leq w(u,y) + w(y,v)$$

for any vertices u, y, v

1) find a MST for G
2) "walk around the tree" forming a path T that visits all the vertices.

3) modify T to take "short-cuts" whenever possible.



T' visits all cities exactly once

$$w(T') \leq w(T)$$

Turns out that

$$w(T') \leq 2 \cdot \text{optimal tour weight.}$$

$\Rightarrow$ this algorithm is a factor 2 approximation