Lecture 20

P and NP

Definition:

$P = \bigcup_{k\geq 1} \mathrm{TIME}(n^k)$

I.e., the set of (decision) problems solvable by computers in *polynomial time*.

$NP = \bigcup_{k\geq 1} \text{Nondeterministic-TIME}(n^k)$

I.e., the set of (decision) problems solvable by computers in *Nondeterministic* polynomial time.

Alternate Definition of NP

A language L is *polynomially verifiable* iff there is a polynomial time procedure v(-,-), (the "verifier") and an integer k such that

for every $x \in L$ there is a "hint" h with $|h| \le |x|^k$ such that v(x,h) = YES and

for every $x \notin L$ there is *no* hint h with $|h| \le |x|^k$ such that v(x,h) = YES

("Hints," sometimes called "certificates," or "witnesses", are just strings.)

Equivalently:

There is some integer k and language L_v in P s.t.:

$$L = \{ x \mid \exists y, |y| \le |x|^k \wedge \langle x,y \rangle \in L_v \}$$

FALSE Example

A_{TM} is in NP

Input: a pair <M,w>

Output: yes/no does M accept w

Hint: y, an accepting computation history of M on w

Clearly, such a y exists for all accepted x and only accepted x, so we accept the right x's and reject the rest.

And it's fast – checking successive configs in the history is at worst quadratic in the length of the history, so the verifier for <x,y> runs in time $|<x,y>|^{O(1)}$.

FALSE Example

A_{TM} is in NP

Input: a pair <M,w>

Output: yes/no does M accept w

Hint: y = 0 or 1, depending on whether M accepts w

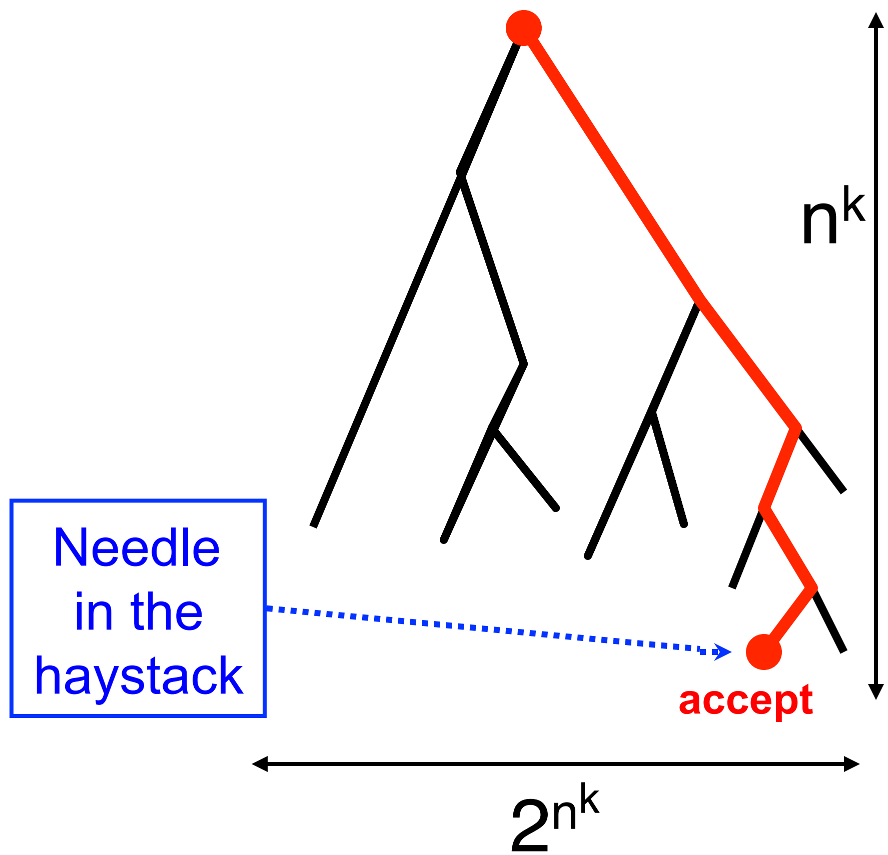
Clearly, such a y exists, so we accept the right x's and reject the rest.

And it's really fast – just read the bit and accept/reject.

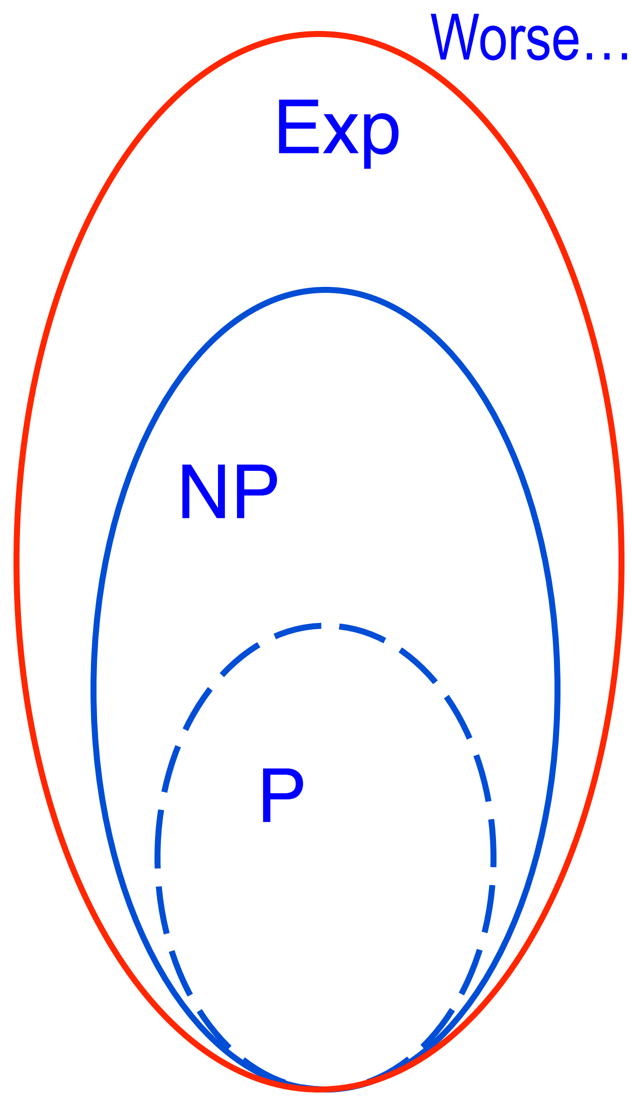
P vs NP vs Exponential Time

Theorem: Every problem in NP can be solved deterministically in exponential time

Proof: "hints" are only n^k long; try all 2^{n^k} possibilities, say by backtracking. If any succeed, say YES; if all fail, say NO.



P and NP



Every problem in P is in NP

- one doesn't even need a hint for problems in P so just ignore any hint you are given

Every problem in NP is in exponential time

I.e., $P \subseteq NP \subseteq Exp$

We know $P \neq Exp$, so either $P \neq NP$, or $NP \neq Exp$ (most likely both)

Problems

Short Path:

4-tuples $\langle G, s, t, k \rangle$, where $G=(V,E)$ is a digraph with vertices s, t, and an integer k, for which there is a path from s to t of length $\leq k$

Long Path:

4-tuples $\langle G, s, t, k \rangle$, where $G=(V,E)$ is a digraph with vertices s, t, and an integer k, for which there is an acyclic path from s to t of length $\geq k$

Short Path

"Is there a short path (< k) from A to B in this graph?"

Any path might work

There are lots of them

I only need one

If I knew one I could describe it succinctly, e.g., "go from A to node 2, then node 42, then ... "

I'd know one if I saw one: "yes, I see there's an edge from A to 2 and from 2 to 42... and the total length is < k"

And if there isn't a short path, I wouldn't be fooled by, e.g., "go from A to node 2, then node 42, then ... "

Long Path

"Is there a long path (> k) from A to B in this graph?"

Any path might work

There are lots of them

I only need one

If I knew one I could describe it succinctly, e.g., "go from A to node 2, then node 42, then ... "

I'd know one if I saw one: "yes, I see there's an edge from A to 2 and from 2 to 42... and the total length is > k"

And if there isn't a long path, I wouldn't be fooled by, e.g., "go from A to node 2, then node 42, then ... "

Mostly Long Paths

“Are the *majority* of paths from A to B long (>k)?”

Any path might work

Yes! → There are lots of the

I only need one

If I knew one I … it
succinctly, e.g. … node
2, then nod…

I'd k… one: "yes, I
see an ed… to 2 and from
2 to 42... and … length > k”

And if there isn’t a long path, I wouldn’t be fooled …

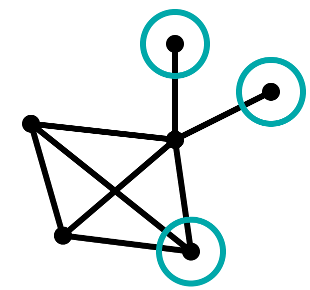
This problem is not believed to be in NP; probably harder

No, this is a *collective* property of the set of all paths in the graph, and no one path overrules the rest

More Problems

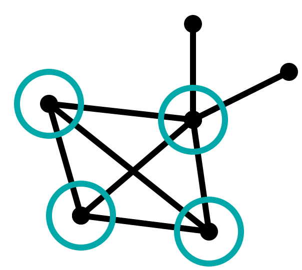
Independent-Set:

Pairs $\langle G,k \rangle$, where G=(V,E) is a graph and k is an integer, for which there is a subset U of V with |U| ≥ k such that no two vertices in U are joined by an edge.



Clique:

Pairs $\langle G,k \rangle$, where G=(V,E) is a graph and k is an integer k, for which there is a subset U of V with |U| ≥ k such that every pair of vertices in U is joined by an edge.



More Problems

Euler Tour:

Graphs G=(V,E) for which there is a cycle traversing each edge once.

Hamilton Tour:

Graphs G=(V,E) for which there is a simple cycle of length |V|, i.e., traversing each vertex once.

TSP:

Pairs $\langle G,k \rangle$, where G=(V,E,w) is a a weighted graph and k is an integer, such that there is a Hamilton tour of G with total weight $\leq k$.

Generic Pattern in These Examples

Set of all x for which *there is a* y with some property P, and

1) y isn't too big ($|y| \le |x|^{O(1)}$), and
2) the property is easy (poly time) to check (given x & y)

"There is a" is a reflection of the quantifier characterization of NP:

L is in NP iff there is some integer k and language L_v in P s.t.:

$$L = \{ x \mid \exists y, |y| \le |x|^k \wedge \langle x,y \rangle \in L_v \}$$

Some similar patterns that suggest problems *not* in NP

Rather than "there is a…" maybe it's "no…" or "for all…"

E.g.

UNSAT: "no assignment satisfies formula," or "for all assignments, formula is false"

Or

NOCLIQUE: "every subset of k vertices is not a k-clique"

These examples are in co-NP: complements of problems in NP. (Quantifier characterization: … $L = \{ x \mid \forall y, |y| \leq |x|^k \wedge \langle x,y \rangle \in L_v \}$ …)

NP =?= co-NP ? Unknown, but seems likely $\neq$.

Some similar patterns that suggest problems *not* in NP

Rather than "there is a…" maybe it's "…is the largest…"

E.g.

MAXCLIQUE: k is the size of the largest clique in G

Or

MINTSP: k is the cost of the cheapest Ham cycle in G

Again, they seem NP-like, but are probably "harder." E.g., not only do you need to prove *existence* of k-clique (a problem in NP) you also need to prove *absence* of a (k+1)-clique (a co-NP question)

Some similar patterns that suggest problems *not* in NP

Rather than "there is a…" maybe it's … something even more complicated, like

the "mostly long paths" example above, or

"there is an exponentially long string y with property P", or

some quantifier structure other than just $\exists$, such as

"$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \forall x_6 \ldots \text{formula}(x_1 \ldots x_n) = \text{True}$"

or many other things

Bottom line:

NP is a *common*, but not *universal*, problem pattern