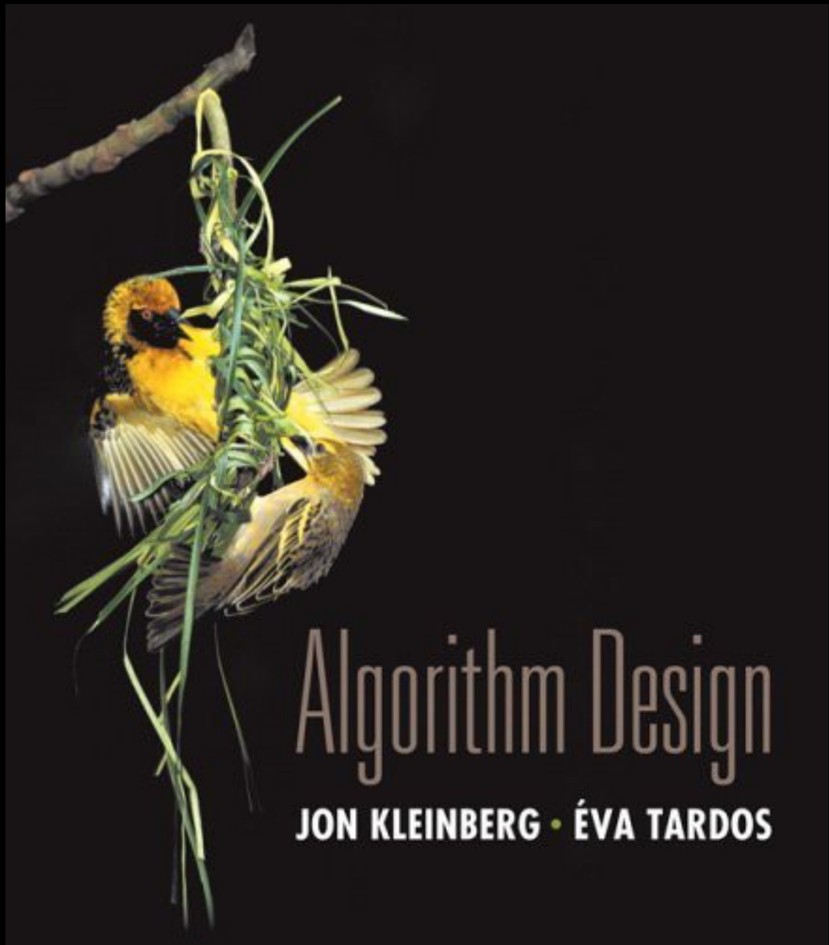


Chapter 8

NP and Computational Intractability

Slides by Kevin Wayne.

8.3 Definition of NP

Decision Problems

Def. Decision problem.

- Σ = Finite Alphabet ; $\Sigma^* \equiv$ {all possible finite strings x of alphabet Σ}
- **A Decision Problem** $X = X \subseteq \Sigma^*$
- **Instance:** any fixed string $s \in \Sigma^*$
- **Question: Does** $s \in X$? (Note: $X \equiv$ { all YES Instances })

Def. Algorithm A solves/decides problem X if for any instance $s \in \Sigma^*$,

$$A(s) = \text{yes} \;\; \textbf{iff} \;\; s \in X.$$

Polynomial time. Algorithm A runs in poly-time **if** for every string s, $A(s)$ terminates in at most $p(|s|)$ "steps", where $p(\cdot)$ is some polynomial, where

$$|s| \equiv \textbf{length} \text{ of } s$$

PRIMES: X = { 2, 3, 5, 7, 11, 13, 17, 23, 29, 31, 37, ... }
Algorithm. [Agrawal-Kayal-Saxena, 2002] $p(|s|) = |s|^8$.

Definition of P

P ≡ { X for which there is a **deterministic poly-time** algorithm }

Problem	Description	Algorithm	Yes	No
MULTIPLE	Is x a multiple of y?	Grade school division	`51, 17`	`51, 16`
RELPRIME	Are x and y relatively prime?	Euclid (300 BCE)	`34, 39`	`34, 51`
PRIMES	Is x prime?	AKS (2002)	`53`	`51`
EDIT-DISTANCE	Is the edit distance between x and y less than 5?	Dynamic programming	`niether neither`	`acgggt ttttta`

The Class NP

Certification algorithm: intuition.

- Certifier views things from "managerial" viewpoint.
- Certifier doesn't determine whether $s \in X$ on its own; rather, it checks a proposed **proof** t that $s \in X$.

Def. Algorithm $C(s, t)$ is a **certifier** for problem X if for every string $s \in \Sigma^*$,

$s \in X$ iff there exists a string t such that $C(s, t) =$ yes.

t = "certificate" or "witness"

Def. NP $\equiv$ { X for which there exists a **poly-time certifier** }

$C(s, t)$ is a poly-time algorithm and $|t| \le p(|s|)$ for some polynomial $p(\cdot)$.

Remark. NP stands for nondeterministic polynomial-time.

Certifiers and Certificates: Composite

COMPOSITES. Given an integer s, is s composite?

Certificate. Any nontrivial **factor** t of s. Note that such a certificate exists **iff** s is composite. Moreover $|t| \le |s|$.

Certifier.

```
boolean C(s, t) {
   if (t ≤1 or t ≥s)
      return false
   else if (s is a multiple of t)
      return true
   else
      return false
}
```

EX. Instance. $s = 437669$.

Certificate. t = 541 or 809.

OBS: $437669 = 541 \times 809$

THM. COMPOSITES is in NP.

Proof. $C(s,t)$ correctly decides if t is a good proof of the fact « s is composite » AND $C(s,t)$ works in time poly(|s|, |t|)

Certifiers and Certificates: 3-Satisfiability

3-SAT.

Instance: CNF formula Φ in the Boolean variable $x_1, x_2, \ldots, x_n$,

Question: is there a **satisfying** assignment $t \in \{0,1\}^n$ for $\Phi(x_1, x_2, \ldots, x_n)$

Certificate. An assignment $t \in \{0,1\}^n$

Certifier. Check that each clause in $\Phi(x_1, x_2, \ldots, x_n)$ has at least one true literal.

Ex.

$$\left(x_1 \vee x_2 \vee x_3\right) \wedge \left(x_1 \vee \overline{x_2} \vee x_3\right) \wedge \left(x_1 \vee x_2 \vee x_4\right) \wedge \left(\overline{x_1} \vee \overline{x_3} \vee \overline{x_4}\right)$$

instance s

$$x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1$$

certificate t

THM. 3-SAT is in NP.

Proof. Homework!

Parameters: n = n. of variables ; m = n. of clauses in $\Phi(x_1, x_2, \ldots, x_n)$

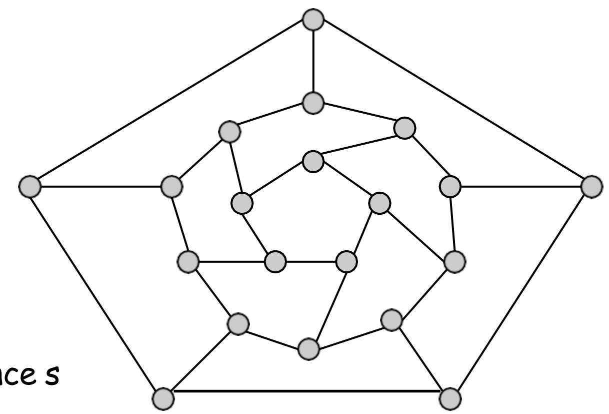
Certifiers and Certificates: Hamiltonian Cycle

HAM-CYCLE.
Instance. an undirected graph $G = (V, E)$ **where** $V = \{1,2,\ldots,n\}$
Question. does there exist a <u>simple</u> cycle **C** that visits <u>every</u> node of V?

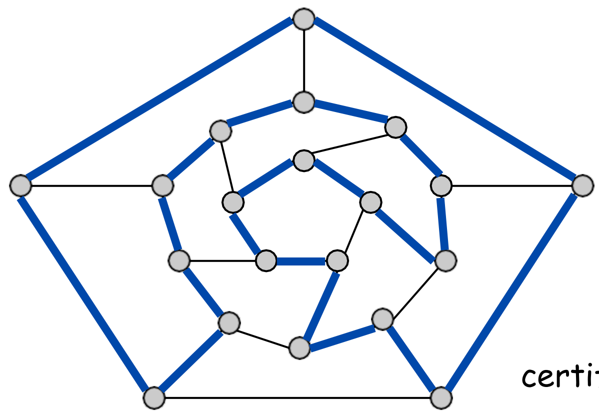
Certificate. A **permutation** Π of $V = \{1,2,\ldots,n\}$.

Certifier $C(G(V,E), \Pi)$. Check that:
1. Π **is a permutation of** V, i.e., **it** contains each $v \in V$ exactly <u>once</u>, and
2. there is **edge** $(v,w) \in E$ between each pair of adjacent nodes **<u,w>** in Π.

Thm. HAM-CYCLE is in NP.
Proof: HOMEWORK!
Parameters: $n = |V|$, $m = |E|$



instance s



certificate t

P, NP, EXP

P. Decision problems for which there is a poly-time algorithm.
EXP. Decision problems for which there is an exponential-time algorithm.
NP. Decision problems for which there is a poly-time certifier.

Claim. $P \subseteq NP$.
Pf. Consider any problem X in P.
- By hyp., there exists a poly-time algorithm $A(s)$ that solves X.
- Certificate: $t = \varepsilon$, certifier $C(s, t) = A(s)$. ▪

Claim. $NP \subseteq EXP$.
Pf. Consider any problem X in NP.
- By definition, there exists a poly-time certifier $C(s, t)$ for X.
- To solve input s, run $C(s, t)$ on all proofs, i.e., strings t with $|t| \le p(|s|)$.
- **How many strings t to check?** if proof's alphabet is binary, then
  # t's $= 2^{p(|s|)}$ **which is ok since we are in EXP**
- Return yes, if $C(s, t)$ returns yes for any of these t's. ▪

AUTO-VALUTAZIONE

- DEF. of DECISION PROBLEM
- DEF. of CERTIFICATE and CERTIFIERS
- DEFINITION OF NP via CERTIFIERS
- HOW TO PROVE THAT A DECISION PROBLEM IS IN NP?
- P vs NP vs EXP