**Complessità Computazionale**

Un algoritmo deve essere efficiente rispettando vincoli di:

* **Tempo**
* **Spazio**
* Accesso alle strutture
* Energia
* Ecc..

**Computabilità**

quali compiti computazionali possono essere risolti?

Non è possibile avere un algoritmo che preveda il termine di un altro, (esempio: non si può prevedere se un algoritmo entrerà in un loop infinito, casi eccezionali esclusi)

**Complessità**

quali compiti computazionali possono essere risolti entro una certa quantità di risorse computazionali?Immagine che contiene testo

Descrizione generata automaticamente

**P=NP**

Ci focalizziamo su problemi decisionali.

Un problema decisionale è un problema che ha come risultato o vero o falso.

Gli input e gli output sono codificati usando un alfabeto (e.g 0,1)

Immagine che contiene diagramma, testo

Descrizione generata automaticamente

Un linguaggio è un insieme di stringhe

**NP**

**Immagine che contiene testo

Descrizione generata automaticamente**

Un algoritmo A è polinomiale nell’input se e solo se per ogni stringa s, A(s) termina in un numero polinomiale di “step” p(|s|), dove p() è polinomiale

|s| è la lunghezza di s

**Gli esponenziali non solo polinomiali**

NP è la classe dei problemi di decisione per i quali esiste un **certificatore** in tempo reale.

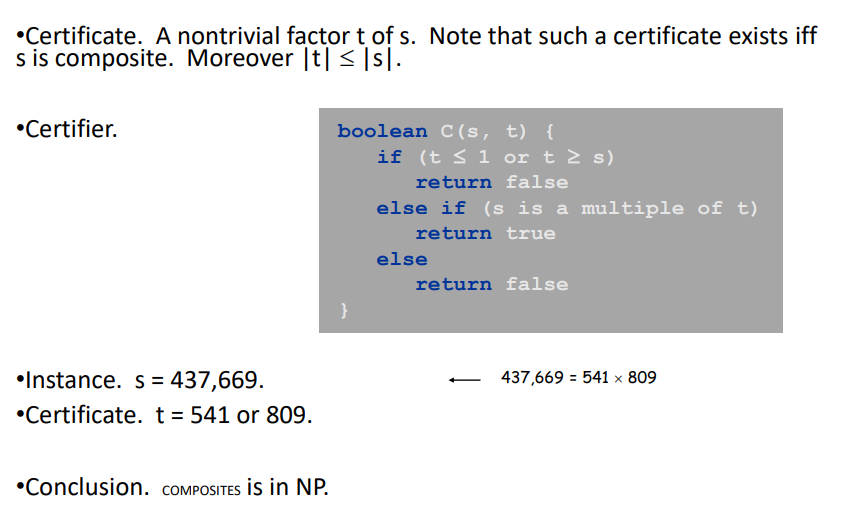
N sta per **non-deterministic**

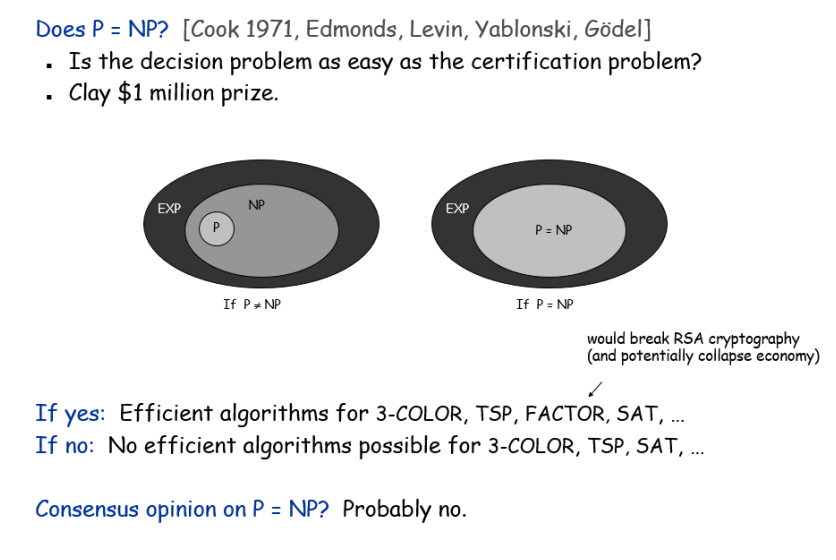
Un certificatore C(s,t) è un algoritmo che ha come input una stringa e a dimostrazione, verifica se una stringa appartiene al dominio del problema in base alla dimostrazione.

Immagine che contiene testo

Descrizione generata automaticamente

•COMPOSITES. Given an integer s, is s composite?





**NP-Complete**

Un problema Y è NP-completo con la proprietà che se posso risolverlo in PTIME allora posso risolvere qualsiasi problema in NP in PTIME