Risposte Complessità:

Siano X e Y problemi di decisione. Si sa che **$X \le_p Y$**. Per ognuna delle seguenti affermazioni dire se essa è: sicuramente vera, falsa, o non si sa:

- Se Y è NP-completo lo è anche X:
  NON SI SA, *affinché X sia NP-completo 1. $X \in NP$ e 2. Per ogni $Z \in NP$, $Z \le_P X$*
- Se X è NP-completo lo è anche Y:
  NON SI SA, *affinché sia vera, $Y \in NP$*
- Se Y è NP-completo e X ∈ NP allora X è NP-completo:
  NON SI SA, *affinché sia vera, per ogni $Z \in NP$, $Z \le_P X$*
- Se X è NP-completo e Y ∈ NP allora Y è NP-completo:
  VERO, *per definizione di NP-completezza.*
- X e Y non possono essere entrambi NP-completi:
  FALSO, *basti pensare che per dimostrare la NP-completezza di Y si sceglie un problema X NP-completo e si dimostra che $X \le_P Y$.*
- Se X è in P, allora Y è in P:
  FALSO, *avere informazioni su X non ci permette di avere ulteriori informazioni su Y.*
- Se Y è in P, allora X è in P:
  VERO. *Infatti, per risolvere un'istanza di X in tempo polinomiale (X in P)*

  1. converto l'istanza di X in un'istanza di Y ($X \le_P Y$)

  2. risolvo l'istanza di Y in tempo polinomiale (Y è in P)

Si considerino 4 problemi A, B, C e D. Ognuno può appartenere o meno alla classe NP. Si conosce l'esistenza delle seguenti riduzioni: **$A \le_P B$, $B \le_P C$, $D \le_P C$**.

- Se A è NP-completo, allora C è NP-completo:
  NON SI SA, *affinché sia vera, anche C deve appartenere a NP (secondo la definizione di NP-completo)*
- A è NP-completo e C ∈ P allora A ∈ P:
  VERO, *dato che esiste una riduzione polinomiale da A a B, possiamo trasformare una istanza di A in una istanza di B in tempo polinomiale. Dato che esiste una riduzione polinomiale da B a C, possiamo trasformare una istanza di B in una istanza di C. Dato che C appartiene a P, possiamo risolvere una istanza di C in tempo polinomiale. Per risolvere una istanza di A, la possiamo trasformare in una istanza di B in tempo polinomiale, possiamo trasformare l'istanza di B in una di C in tempo polinomiale e possiamo risolvere l'istanza di C in tempo polinomiale. Quindi $A \in P$.*
- Se A è NP-completo e B ∈ NP, allora B è NP-completo:
  VERO, *se A è NP-completo, per definizione di NP-completezza, ogni problema in NP e riducibile ad A. Dato che $A \le_P B$, per transitività, ogni problema in NP si può ridurre a un problema di B. Inoltre, $B \in NP$, quindi (per definizione), B è NP-completo.*
- Se C è NP-completo allora D ∈ NP:
  VERO *perché*
  a. dato che $D \le_P C$, un'istanza di D può essere trasformata in un'istanza di C in tempo polinomiale. Dato che C è NP-completo, C è in NP, quindi esiste un validatore per C. L'algoritmo per validare D può prevedere la conversione dell'istanza da D a C e la validazione di C. Quindi D appartiene a NP.
  b. 1. se D è anche in P, comunque è in NP, 2. dato che si riduce a C, D non può essere più complesso di C.

Mostrare in maniera formale e rigorosa le seguenti inclusioni tra le classi di complessità, enunciando in maniera precisa eventuali risultati intermedi.

- P ⊆ NP:
  Consideriamo un qualsiasi problema L in P. Per definizione, esiste un algoritmo M che decide L in tempo polinomiale. Considero l'algoritmo di verifica V che sull'input y:

  - se $y \neq \langle w,\varepsilon \rangle$ con w stringa e ε certificato, rifiuta y

  - se $y = \langle w, \varepsilon \rangle$ con w stringa e ε certificato, simula M su $\langle w \rangle$ e accetta y se e solo se M accetta $\langle w \rangle$.
  V verifica L in tempo polinomiale.
- NP ⊆ EXP:
  Consideriamo un qualsiasi problema L in NP. Per definizione, esiste un certificatore polinomiale C(s,t) per L. Per risolvere L su un input s, definisco il seguente algoritmo per decidere L:
  - genero tutte le possibili stringhe t con $|t| <= p(|s|)$ e verifico C(s,t).
  - se il certificatore restituisce SI per qualche certificato t, allora posso restituire SI.
  Il certificatore ha tempo polinomiale mentre l'algoritmo che risolve L ha tempo esponenziale.
- P ⊆ co-NP:
  co-NP = $\{L \mid \bar{L} \in NP\}$, $L \in P$ => $\bar{L} \in P$ => $\bar{L} \in NP$ => $L \in$ co-NP => $P \subseteq$ co-NP
  P è chiuso rispetto al complemento; $P \subseteq NP$; per def. di co-NP

Considerando le classi di problemi: decidibili, P, NP, NP-COMPLETI e EXP, disegnare un diagramma che mostra le relazioni tra queste classi sotto l'ipotesi che P è diverso da NP.

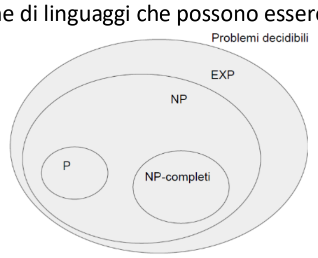
Problema decidibile= esiste un decider che riconosce il linguaggio associato al problema decidibile.

P= insieme dei linguaggi che possono essere decisi da una MdT deterministica a un solo nastro in tempo polinomiale.

NP= insieme di linguaggi che possono essere decisi da una MdT non deterministica in tempo polinomiale (o che possono essere verificati in tempo polinomiale).

NP-completi= problemi in NP e tali che qualsiasi problema in NP si può ridurre polinomialmente a lui.

EXP= insieme di linguaggi che possono essere decisi da una MdT deterministica in tempo esponenziale.



P ⊂ NP dato che P è diverso da NP, per hp

P ∩ NP-COMPLETO = Ø perché P è diverso da NP

NP-COMPLETO ⊂ NP perché ha vincoli più stringenti

NP ⊂ EXP perché possiamo risolvere un problema in NP generando ogni input e usando il suo verificatore. La generazione di tutti gli input prende tempo esponenziale.