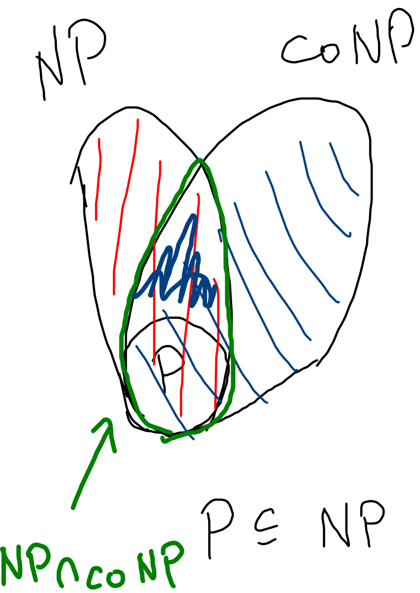
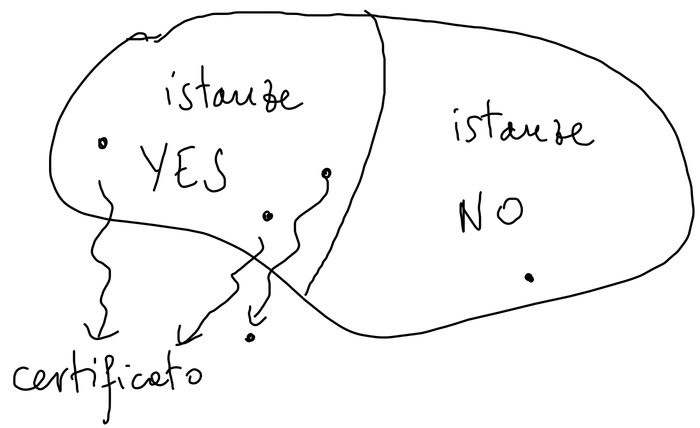
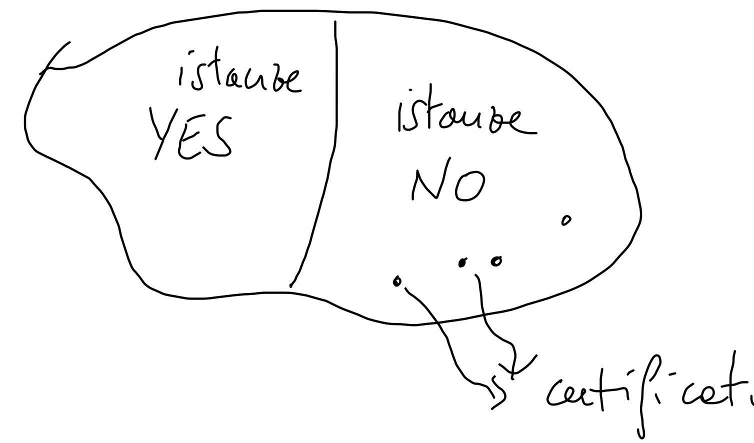
$X \in NP$ $\qquad\qquad$ $Y \in co\text{-}NP$

Insieme delle istanze di X







$P \subseteq NP$, $P \subseteq co\text{-}NP \implies P \subseteq NP \cap coNP$

$X \longrightarrow \overline{X} :=$ I è istanza YES di $\overline{X}$

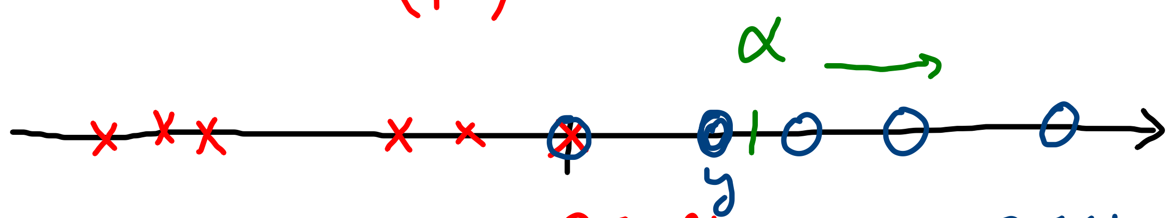
$\iff$ I è istanza NO di X

$X \in NP \iff \overline{X} \in co\text{-}NP$

valore del flusso

$\xrightarrow{\hspace{3cm}}$

(P)



$\underset{\text{MAX}}{\text{OPT}}(\text{flusso}) = \underset{\text{MIN}}{\text{OPT}}(\text{taglio})$

$\xrightarrow{\hspace{3cm}}$

Capacità dei tagli

(D)

VERTEX COVER:
Dato un grafo G, e dato un intero k, decidere se G ha un vertex cover di cardinalità $\leq k$.

VERTEX COVER $\in NP$ e vale (2)

$\Downarrow$

VERTEX COVER è NP-completo

Problemi di decisione

$X \in$ <u>NP-completo</u> $\iff$

$$\begin{cases} (1)\ X \in NP \\ (2)\ \forall Y \in NP,\ Y \leq_p X \\ (0)\ X \text{ è un problema di decisione} \end{cases}$$

NP è una classe di problemi di decisione

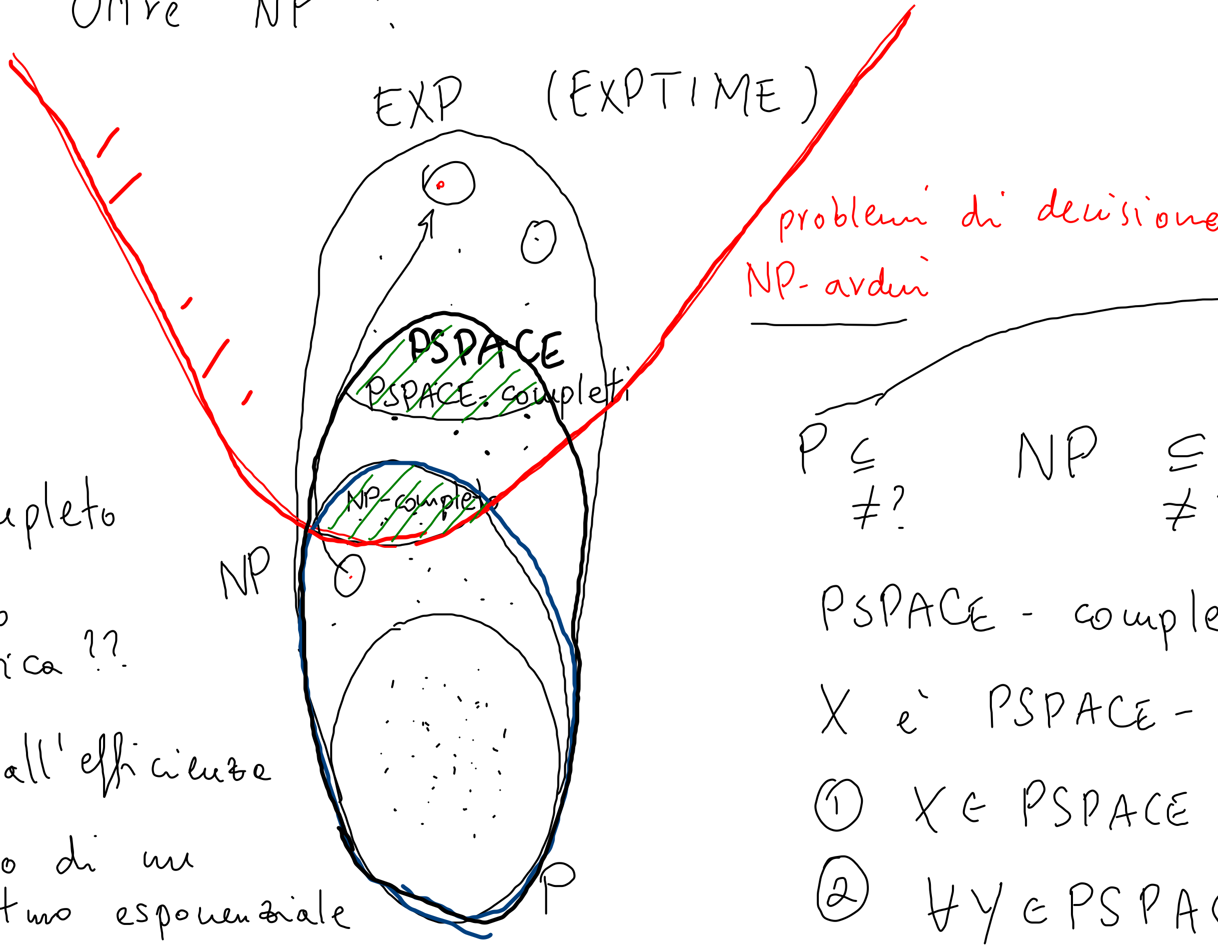
X è NP-arduo $\overset{def}{\iff}$ $\forall Y \in NP,\ Y \leq_p X$

Può essere anche un problema di ricerca o di ottimizzazione

Es. MIN VERTEX COVER: $\rightarrow$ è NP-arduo

Dato un grafo G, trovare un vertex cover di G di cardinalità <u>minima</u>.

Oltre NP?



(2) Rinunciare all'ottimalità

↳ Algoritmi approssimati (approssimazione dell'ottimo)

(3) Concentrarsi su casi particolari

X ∈ NP-completo

Cosa faccio in pratica??

(1) Rinuncio all'efficienza

mi accontento di un algoritmo esponenziale

k^n $\quad$ 1.001^n

$P \underset{\neq ?}{\subseteq} NP \underset{\neq ?}{\subseteq} PSPACE \underset{\neq ?}{\subseteq} EXP$ $\quad$ ($P \neq EXP$)

PSPACE-completi:

X è PSPACE-completo se:

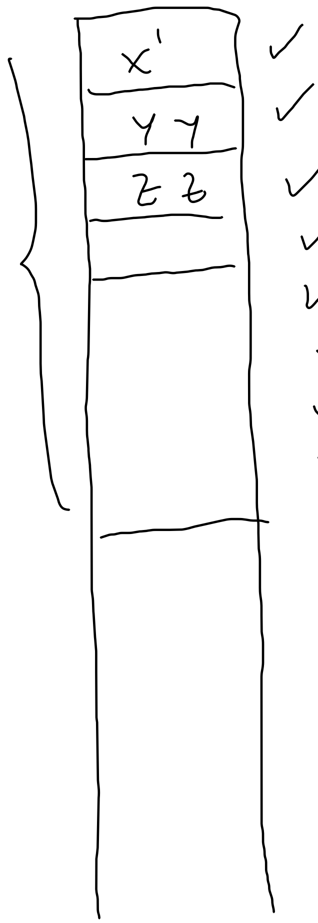
(1) $X \in PSPACE$

(2) $\forall Y \in \underline{PSPACE} \leq_p X$

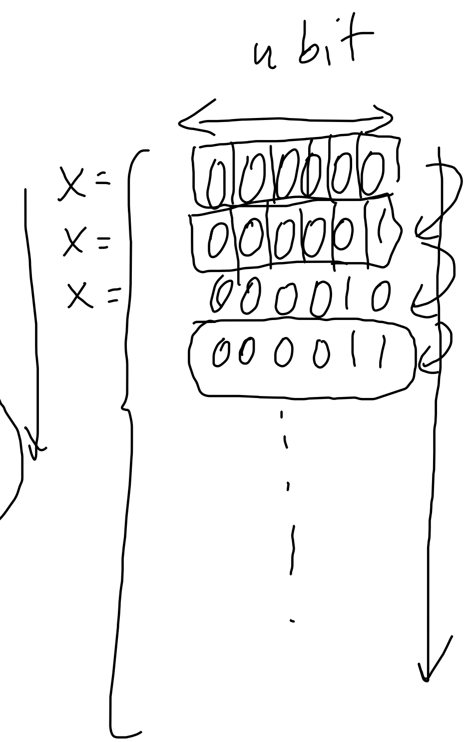
$2^{k \cdot \log n}$

Vertex Cover: esiste un algoritmo tempo $O(2^k \cdot k \cdot n)$ (molto meglio della forza bruta: $\binom{n}{k}$: $O(n^k)$)

In $T(n)$ passi, posso utilizzare al più $T(n)$ celle di memorie



A = new int[100'000]



$T(n) = 2^n$

Spazio utilizzato

$S(n) = n$