ESERCIZI.

22/12

*) Mostrare che se P = NP, allora NP = NP-COMPLETE per ogni linguaggio diverso da $\emptyset$ e Σ^*.

Sappiamo che NP-COMPLETE $\subseteq$ NP. Mostriamo che NP $\subseteq$ NP-COMPLETE.

Sia $A \in NP$, devo far vedere che per ogni $L \in NP$ si ha $L \leq_m^P A$. Sappiamo $A \neq \emptyset, \Sigma^*$, per cui esistono x_{yes}, x_{no} t.c. $x_{yes} \in A$ e $x_{no} \notin A$.

La riduzione: deterministica (P = NP)

- Dato x, usa la TM per decidere $x \in L$.
- Se M accetta, ritorna x_{yes}. Altrimenti x_{no}.

($\Rightarrow$) Se $x \in L$, $M(x)$ accetta col $f(x) = x_{yes} \in A$.

($\Leftarrow$) Se $x \notin L$, $M(x)$ rifiuta col $f(x) = x_{no} \notin A$.

*) Dare una TM che decide $L = \{0^k 1^k : k \geq 0\}$ in tempo $O(n^2)$

Quella che abbiamo visto a lezione:

- Scorriamo il nastro e controlla sia della forma $0^* 1^*$. Tempo: $O(n)$.
- Torna indietro e avvia loop fino a che il nastro contiene almeno uno 0 o almeno un 1:
  - Cambia uno 0 con X e un 1 con X

- Se il nastro contiene solo X accetta. Altrimenti rifiuta. Tempo: $O(n)$

Tempo loop: $2n$ passi $= O(n)$ e viene ripetuto $n/2$ volte. Ovvero $O(n^2)$ passi.

Tempo: $O(n) + O(n^2) + O(n) = O(n^2)$.

*) Stesso linguaggio in $O(n \log n)$ tempo.

L'idea: togliere più 0 e 1 contemporanea-mente.

```
0000000000000 1111111111111
X0X0X0X0X0X0XX1X1X1X1X1X1X
XXX0XXX0XXX0XXXX1XXX1XXX1X
XXXXXXX0XXXXXXXXXXX1XXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXX
```

- Controllo formato 0^*1^*. Tempo: $O(n)$.
- Conto lunghezza input; se dispari rifiuto. Tempo: $O(n)$.
- Per il loop: Ripeto fino a che c'è almeno uno 0 o almeno un 1.
  - Sostituisco metà degli 0/1 con X. Come nella figura (1 sì e 1 no). Tempo: $O(n)$.
- Controllo tutte XX..X. Tempo $O(n)$.

Tempo totale: $O(n \log n)$.

*) Stesso linguaggio, tempo $O(n)$ con 2 nastri.

- Copio gli 0 sul secondo nastro. La testina su input punta al primo 1. $O(n)$.
- Riposiziono la testina su secondo nastro all'inizio. $O(n)$.
- Scorriamo i due nastri e controllo che raggiungano "⊔" allo stesso momento. $O(n)$.

*) Mostrare che se $P=NP$ esiste algoritmo polinomiale che dato un grafo restituisca una CLIQUE massimale nel grafo.

Ci ricordiamo che K-CLIQUE è in NP.
Per $K \in \mathbb{N}$. Quindi K-CLIQUE è in P:
$\exists$ TM che lo decide in tempo polinomiale.

- Calcolo la dimensione della clique massima usando $M(<G,1>)$, $M(<G,2>)$, ...
Quando M rifiuta per $K+1$, la dimensione è $K \in \mathbb{N}$.
- Calcolo la clique. Per ogni $v \in V$ lo rimuovo, ottengo G' e calcolo la dimensione della CLIQUE MASSIMA. Se è diminuita v fa parte della CLIQUE in G e quindi lo reinserisco.
- Alla fine mi rimangono solo nodi della CLIQUE.

*) Mostrare che NP è chiusa rispetto a *.
Se $A \in NP$ esiste NTM N che decide A in tempo polinomiale. Costruisco NTM N' che deci-

$A' = A^*$:

- Su input x lo divido non deterministicamente in $x = x_1 x_2 \dots x_k$
- Controllo $x_i \in A$ usando N.
- Se N su un ramo di computazione N accetta sempre, allora accetto.

*) Mostrare $DTIME(2^n) = DTIME(2^{n+1})$ e $DTIME(2^n) \subsetneq DTIME(2^{2n})$.

La prima è banale: $2^{n+1} = 2 \cdot 2^n = O(2^n)$.

Per la seconda: 2^{2n} è tempo costruttibile, basta dare in output $10 \dots 0$. Per il teorema di gerarchia di tempo esiste A decidibile in tempo $O(2^{2n})$ ma non in tempo:

$$O\left(\frac{2^{2n}}{\log_2 2^{2n}} \right) = O\left(2^{2n} / 2n \right)$$

$2^n = o\left(2^{2n} / 2n \right)$ ovvero $A \notin DTIME(2^n)$.

*) $NTIME(n) \subsetneq PSPACE$.

Come faccio: $NTIME(n) \subseteq NSPACE(n)$ (il tempo limita lo spazio). Poi

$$NSPACE(n) \subseteq SPACE(n^2)$$

per Savitch. Infine, $SPACE(n^2) \subsetneq SPACE(n^3)$ per Teorema di gerarchia. $SPACE(n^3) \subseteq PSPACE$.

*) Sia $pad : \Sigma^* \times \mathbb{N} \to \Sigma^* \#^*$ tale che

$pad(s, l) = s\#^i \quad i = \max(0, l - |s|)$.

$\forall k \in \mathbb{N}$, $A \in P$ sse $pad(A, n^k) =$

$= \{ pad(x, n^k) : x \in A \} \in P$

Se $A \in P$, allora $pad(A, n^k) \in P$ perché posso calcolare $w \in pad(A, n^k)$ scrivendo $w = s\#^l$ per s senza $\#$ e con $|w| = n^k$, e posso verificare $s \in A$.

Se $pad(A, n^k) \in P$, allora posso decidere se $w \in A$ facendo padding di w con tanti $\#$ fino a lunghezza n^k.

*) $P \neq SPACE(n)$.

Sia $P = SPACE(n)$ e sia $A \in SPACE(n^2)$ ma $A \notin SPACE(n)$. A esiste per il Teorema di gerarchia.

Consideriamo: $pad(A, n^2)$ è in $SPACE(n)$

perché ho spazio a sufficienza per lanciare la TM che spazio n^2 per decidere A.
Ovvero $pad(A, n^2) \in SPACE(n) = P$.
Ma allora $A \in P = SPACE(n)$. Impossibile.

*) Mostrare che:

- Se $L_1, L_2 \in coNP$, allora $L_1 \cap L_2 \in coNP$.
- Se $L \in NP$, $L_1 \subsetneq L$ ed $L_1 \in coNP$, allora $L \setminus L_1 \in NP$.

Siccome $L_1, L_2 \in coNP$, $\overline{L_1}, \overline{L_2} \in NP$. Ovvero $\overline{L_1} \cup \overline{L_2} \in NP$. $L_1 \cap L_2 = \overline{(\overline{L_1} \cup \overline{L_2})} \in coNP$

Se $L_1 \in coNP$, allora $\overline{L_1} \in NP$. NP è chiuso per intersezione: $L \cap \overline{L_1} = L \setminus L_1 \in NP$.