ALGORITMI - Complessità	**Anno Accademico 2014-15**
Prova in Itinere 1	

Nome e Cognome: Matricola:

Importante: Usare solo i fogli forniti dal docente, riportando Nome Cognome e Matricola su ogni foglio. Scrivere in modo **leggibile** e **giustificare** ogni affermazione. Risposte non giustificate saranno valutate **0**

Si chiede di risolvere 3 esercizi:
(i) uno a scelta tra 1 e 2,
(ii) uno a scelta tra 3 e 4,
(iii) almeno una delle domande di 5 oppure l'esercizio 6.

Esercizio 1 - 8 punti

Una delle definizioni date a lezione per $\mathcal{NP}$ è in termini di proprietà esprimibili come predicati logici:

$$\mathcal{NP} = \{A(x) = \exists w : B(x,w),\ B \in \mathcal{P} \text{ e } |w| = O(poly(|x|))\}$$

Secondo la stessa prospettiva consideriamo la seguente classe definita in termini logici

$$\mathcal{OP} = \{A(x) = \exists w : B(x,w),\ B \in \mathcal{P} \text{ e } |w| = O(log(|x|))\}.$$

Si dimostri che $\mathcal{OP} = \text{co}-\mathcal{OP}$.

Esercizio 2 - 8 punti

Un nostro amico è un pò confuso circa la definizione di co-$\mathcal{NP}$. Ci ha chiesto di chiarirgli se le seguenti affermazioni sono vere o false

1. se $\mathcal{P} \neq \text{co}-\mathcal{NP}$ allora 3-Sat $\not\leq$ 2-Sat (non può esistere una riduzione da 3-Sat a 2-Sat).
2. Possiamo usare il seguente algoritmo non deterministico per risolvere Tautologia

   > L'algoritmo usando l'istruzione non-deterministica **goto both** crea tutti gli assegnamenti per la formula ϕ da verificare. Quindi per ogni assegnamento esiste un ramo di computazione. Ogni ramo di computazione verifica se l'assegnamento creato soddisfa ϕ ed in tal caso dà in output SI e termina, altrimenti termina senza fornire output.

   Il nostro amico è convinto che questo algoritmo funzioni e si chiede come mai esso non dimostra che $\mathcal{NP} = \text{co-}\mathcal{NP}$.

Esercizio 3 - 12 punti

Si consideri la seguente variante di 3-SAT:

MAJORITY 3-SAT (MAJ-3-SAT)
Input: una formula CNF ϕ in cui ogni clausola ha 3 letterali
Output: *yes* se e solo se esiste un assegnamento che rende veri almeno due letterali in ogni clausola

Quale delle due seguenti affermazioni vale? (in parentesi una possibile strada per provarle)

- MAJ-3-SAT è in $\mathcal{P}$ (mostrare una riduzione usando 2-SAT).
- MAJ-3-SAT è $\mathcal{NP}$-completo (mostrare una riduzione usando 3-SAT).

Esercizio 4 - 12 punti

NSA sostiene di essere capace di individuare i gruppi terroristici controllando le interconnessioni tra le persone che spia. Dicono che prima di un attacco terroristico ogni partecipante contatta o viene contattato da ogni altro partecipante all'attacco. NSA pensa che sia quindi giusto studiare il seguente problema:

MASSIMA INTERCONNESSIONE
Input: n individui, l'insieme di coppie di individui che hanno comunicato tra loro, e un numero $k \leq n$
Output: *yes* se e solo se esiste un gruppo C di almeno k individui tale che ogni individuo di C ha comunicato con ogni altro individuo di C

Provare che il problema MASSIMA INTERCONNESSIONE è $\mathcal{NP}$-completo. (Una possibile riduzione è da INDEPENDENT SET).

Esercizio 5 - 10 punti

Supponiamo di voler controllare se in una stringa di parentesi aperte e chiuse di 2 possibili tipi: “ [” e “ (”, le parentesi risultano correttamente accoppiate e annidate.

Per esempio la stringa [ ] [( )] [[ ] ( )] è corretta, mentre la stringa [[([ )]]] e la stringa [( ) [ ] non sono corrette (nella prima gli annidamenti sono sbagliati, nella seconda manca una parentesi chiusa).

(i) Mostrare che il problema è in L.

(ii) Mostrare che se aggiungiamo la limitazione che l'input può essere letto solo da sinistra a destra allora ogni algoritmo che risolve il problema richiede $\Theta(n)$ bit di memoria (dove n è la lunghezza della stringa input)

Esercizio 6 - 10 punti

A lezione abbiamo provato che se $\mathcal{P} = \mathcal{NP}$ allora vale anche EXP = NEXP.

Provare che per una qualsiasi costante $c > 0$ se $\text{NTIME}(2^{2^{O(n^c)}}) \neq \text{TIME}(\ 2^{2^{O(n^c)}})$ allora $\mathcal{P} \neq \mathcal{NP}$.