# Introduzione agli Algoritmi (secondo canale, A.A. 2010-2011)
# Prova scritta del 20 Giugno 2011
# Prof.ssa Irene Finocchi

TEMPO CONCESSO: 1H30M PER CIASCUNA PARTE.
CONSEGNATE UN FOGLIO PER ESERCIZIO, PENA L'ANNULLAMENTO DEL COMPITO.
COME INTESTAZIONE DI CIASCUN FOGLIO, SCRIVETE NOME, COGNOME, E NUMERO DI ESERCIZIO. NEL CASO IN CUI NON RIUSCIATE A SVOLGERE UN ESERCIZIO, CONSEGNATE COMUNQUE UN FOGLIO CONTENETE L'INTESTAZIONE, MA SENZA SOLUZIONE.
E' AMMESSO L'USO DI LIBRI ED APPUNTI, MA NON DI PORTATILI E TELEFONI CELLULARI.

PARTE I

**Esercizio 1**

Sia *A[1;n]* un array contenente il *grado di popolarità* degli $n$ amici di una persona $P$. Il grado di popolarità di una persona è definito come il più grande numero $h$ tale che quella persona ha almeno $h$ amici ciascuno con grado di popolarità $\geq h$. L'array $A$ quindi contiene $n$ interi non-negativi.

Si progetti un algoritmo che calcoli il grado di popolarità di $P$, ovvero il valore:

$$h = \max_{1 \leq i \leq n} \{ A[i] \text{ tale che } P \text{ ha almeno } A[i] \text{ amici con grado di popolarità} \geq A[i] \}$$

Si analizzino la correttezza e il tempo di esecuzione dell'algoritmo proposto, che dovrebbe essere $O(n \lg n)$.

**Esercizio 2**

Si risolva la seguente relazione di ricorrenza:

$$T(1) = 1$$
$$T(n) = 3T(n/2) + n^2 \quad \text{se } n>1$$

# Introduzione agli Algoritmi (secondo canale, A.A. 2010-2011)
# Prova scritta del 20 Giugno 2011
# Prof.ssa Irene Finocchi

PARTE II

**Esercizio 3**

a) Siano dati un maxHeap H e due elementi $x \in H$ e $y \in H$. Siano $h_x$ e $h_y$ le altezze dei sottoalberi di H radicati in x e y. Dimostrare o confutare con un controesempio la seguente affermazione: se $x>y$ allora $h_x>h_y$.

b) Sia H un maxHeap. Dimostrare o confutare con un controesempio la seguente affermazione: per ogni livello *t*, la somma delle chiavi dei nodi nel livello *t* è maggiore o uguale alla somma delle chiavi dei nodi nel livello *t+1*.

c) Potremmo rilassare la definizione di albero AVL richiedendo che tutti i nodi, tranne al più tre di essi, abbiano fattore di bilanciamento $\in [-1,1]$ e che i tre nodi speciali abbiano fattore di bilanciamento $\in [-5,5]$. Dimostrare o confutare con un controesempio la seguente affermazione: l'altezza di un albero AVL rilassato con *n* nodi è ancora $O(\log n)$.

**Esercizio 4**

Dato un albero binario di ricerca *T* rappresentato tramite puntatori ai figli e data una chiave $x \notin T$, progettare un algoritmo che, presi in input *T* e *x*, restituisce la coppia di radici $(T_1,T_2)$ di due *nuovi* alberi binari di ricerca, il primo contenente gli elementi di T $<x$ e il secondo contenente gli elementi di T $>x$. L'algoritmo *non* deve modificare l'albero di input T e *la struttura di $T_1$ e $T_2$ deve essere conforme a quella di T*: ovvero, se un nodo *u* è antenato di un nodo *v* in *T*, lo deve essere anche in $T_1$ o $T_2$ (nel caso in cui *u* e *v* appartengano entrambi allo stesso albero di output).