

# INFORMATICA

1. Il candidato svolga due esercizi a scelta dei quattro seguenti:

- (a) Il punto  $i$  del piano cartesiano, di coordinate  $(x_i, y_i)$  si dice che *domina* il punto  $j$ , di coordinate  $(x_j, y_j)$ , se e solo se  $x_i \geq x_j$ , e  $y_i \geq y_j$ . Scrivere un algoritmo che, ricevuti in ingresso  $n$  punti distinti del piano cartesiano (rappresentati mediante le loro coordinate), stabilisce quali di loro non e' dominato da nessun altro punto dell'insieme. Descrivere le strutture dati utilizzate, e indicare la complessita' dell'algoritmo, giustificando il valore indicato.
- (b) Descrivere la gestione della memoria nei sistemi operativi che utilizzano la paginazione, illustrando le problematiche che emergono e le soluzioni adottate. Si consideri poi un computer avente una memoria associativa (relativamente) grande rispetto alla memoria principale, e altri due livelli di memoria secondaria: un hard disk e una memoria di massa piu' grande ma piu' lenta (ad esempio, i dischi di utenti peer nei sistemi peer-to-peer di internet, da cui si possono scaricare files condivisi). Indicare come e' conveniente organizzare la gestione della memoria con paginazione dinamica in questo caso.
- (c) La definizione dei linguaggi di programmazione con grammatiche libere non dà conto di vincoli che sono espressi informalmente, ad esempio il riferimento a una variabile deve essere preceduto da una dichiarazione della variabile stessa. Si dia almeno un altro esempio motivando formalmente l'impossibilità di esprimere i vincoli con una grammatica libera.
- (d) Dijkstra propose il linguaggio dei *guarded commands* con la seguente sintassi:

$$c ::= \text{skip} \mid \text{abort} \mid X := e \mid c_1; c_2 \mid \text{if } gc \text{ fi} \mid \text{do } gc \text{ od}$$
$$gc ::= b_1 \rightarrow c_1 + b_2 \rightarrow c_2$$

dove  $e$  è un'espressione aritmetica, e  $gc$  un guarded command.

Le espressioni booleane  $b_1$  e  $b_2$  sono chiamate guardie, e  $c_1$  e  $c_2$  sono comandi. L'esecuzione di *skip* non cambia lo stato. L'esecuzione di *abort* non porta ad uno stato finale. Se nessuna guardia valuta a vero in uno stato, allora il guarded command fallisce, ossia non dà uno stato finale, altrimenti esegue uno dei comandi la cui guardia è vera. Il comando *if gc fi* si comporta come  $gc$  se  $gc$  non fallisce, altrimenti si comporta come *abort*. Il comando *do gc od* esegue ripetutamente  $gc$  fino a quando  $gc$  non fallisce, termina quando  $gc$  fallisce.

Supponiamo che esista una funzione di valutazione per espressioni aritmetiche e booleane, e che questa termini sempre, senza effetti collaterali.

Dare una semantica formale al linguaggio.

2. Si illustri un'area di ricerca innovativa in informatica, discutendone risultati raggiunti e possibili sviluppi futuri