

# Laboratorio di programmazione e Informatica 1

## - A.A. 2018-2019 -

Terzo appello - 16 settembre 2019

### ESERCIZI

Tempo a disposizione: 2 ore e mezza. Ogni esercizio vale 10 punti. Per superare l'esame bisogna raggiungere un minimo di 5 punti su ogni esercizio e un minimo totale di 18 punti.

#### Esercizio 1

Si progetti una funzione in C che, data una lista  $P$  di punti del piano cartesiano, e data una retta  $r$  tramite l'equazione  $y = mx + p$  con  $m > 0$ , estrae i punti di  $P$  che si trovano esattamente sulla retta o al di sopra di essa e li mette in una nuova lista  $H$ . Alla fine la lista  $P$  dovrà contenere i punti che si trovano al di sotto della retta  $r$ .

1. Si definisca il tipo nodo lista adatto per la lista  $P$ .
2. Si implementi in C una funzione denominata `SpezzaSemipiani`, che prende in input la lista di punti  $P$  e i parametri  $m$  e  $q$  della retta  $r$  ed estrae la lista  $H$  come spiegato in precedenza. La funzione non deve creare nuovi nodi ma deve utilizzare quelli della lista di input ed inoltre deve essere preservato l'ordinamento dei nodi rispetto alla lista originale.

#### Esercizio 2

Considerare il seguente problema: dato un vettore  $v$  di interi, trovare il numero di somme di coppie di elementi distinti che restituiscono un certo valore  $m$ . Esempio Se il vettore fosse  $v = \{1, 7, 4, 5, 2, 3\}$  ed  $m = 8$ , ci sono 2 coppie che danno come somma 8, cioè  $1+7$  e  $5+3$ . Non va considerato il fatto che  $4+4=8$  (non si tratta di una somma di elementi distinti).

- Scrivere una funzione C di prototipo `int countSums(int v[], int n, int m)`, dove  $n$  è la lunghezza del vettore  $v$ , che restituisce il numero delle coppie distinte di  $v$  di somma  $m$ .
- Sotto la precondizione che il vettore  $v$  sia strettamente crescente, scrivere una funzione lineare in  $n$ .
- Se al punto precedente è stata data una funzione iterativa, scrivere una funzione ricorsiva (o viceversa).
- Argomentare la correttezza e la complessità delle funzioni proposte ai punti precedenti (possibilmente aiutandosi con le condizioni logiche sulle proprietà soddisfatte durante l'esecuzione dei programmi).

#### Esercizio 3

Dare la definizione di *albero binario* precisando il significato dei termini *radice*, *nodo interno*, *foglia*. Scrivere una struttura di puntatori per descrivere i nodi di un albero binario e dare una funzione ricorsiva che prende in input un albero binario tramite un puntatore alla radice e conta quante foglie ci sono nell'albero.