

Classifica del torneo

I risultati di un torneo di pallavolo a eliminazione diretta vengono rappresentati nel sistema informatico della federazione sportiva come un albero binario di partite. Ogni nodo dell'albero rappresenta una partita e contiene le seguenti informazioni:

- **teamId**: l'identificativo della squadra che ha vinto la partita (di tipo intero)
- **setWon**: il numero di set vinti durante la partita dalla squadra vincitrice (di tipo intero)
- **setLost**: il numero di set persi durante la partita dalla squadra vincitrice (di tipo intero)

La classifica del torneo viene calcolata assegnando un punteggio a ciascuna squadra in base ai risultati ottenuti. Di seguito viene riportato lo pseudocodice che descrive come vengono calcolati i punteggi delle squadre:

Algorithm 1 `getScores(root, scores, m)`

Input: *root* la radice di un albero binario, *scores* un array di numeri a virgola mobile di lunghezza *n* tale che $s_i = 0$ e $i = 0, \dots, n$ dove s_i è l'elemento *i*-esimo dell'array *scores*, *m* un numero intero.

Output: Nessuno. **Side effect:** *scores* contiene il punteggio di ogni squadra.

if *root* = nil **then**

return

end if

scores[*root*.teamId] \leftarrow *scores*[*root*.teamId] + (*root*.setWon - *root*.setLost) \times *m*

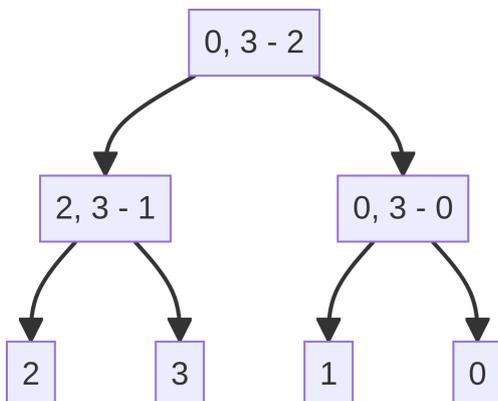
getScores(*root*.left, *scores*, $\frac{m}{2}$)

getScores(*root*.right, *scores*, $\frac{m}{2}$)

Precondizioni: si assuma che tutti gli elementi di *scores* siano inizializzati a 0 ed $m \in \mathbb{N} \setminus \{0\}$.

Una volta ottenuto l'array con i punteggi delle squadre si può facilmente trovare la squadra vincitrice con un algoritmo che abbia complessità $O(n)$, dove *n* è la lunghezza dell'array *scores*.

Si consideri come esempio un torneo con 4 squadre, identificate dai numeri da 0 a 3. Le foglie dell'albero rappresentano solamente i nomi delle squadre (i valori **setWon** e **setLost** sono inizializzati a 0). I nodi genitori contengono nel campo **teamId** l'identificativo della squadra vincente e nei campi **setWon** e **setLost** il numero di set vinti e persi rispettivamente. L'albero binario di partite è il seguente:



Una rappresentazione testuale dell'albero, più adatta ad essere stampata nel terminale, è la seguente:

0, 3 - 2
0, 3 - 0
0
1
2, 3 - 1
2
3

Si chiede di implementare in linguaggio C le funzioni:

- `void getScores(Node *root, double *scores, double m)`
- `int getWinner(double *scores, int n)`

la funzione `getScores` deve calcolare i punteggi delle squadre partendo dalla radice dell'albero binario di partite, mentre la funzione `getWinner` deve restituire l'id della squadra vincente.

Se necessario, si possono aggiungere funzioni di supporto, ma è necessario che le funzioni richieste siano implementate esattamente con i prototipi forniti.

Lo pseudocodice non è codice completo in C: non considera i tipi dei vari oggetti, l'uso di puntatori/riferimenti, e dunque dell'operatore “->” anziché l'operatore “.”: ‘e lasciato allo studente il compito di determinare questi dettagli come adeguato.

i Note

L'implementazione dell'albero in C è una semplice variante di quella utilizzata durante le lezioni del corso e viene fornita già implementata nei file dell'esercizio come segue:

```
typedef struct node
{
    int teamId;
    int setWon;
    int setLost;
    struct node *left;
    struct node *right;
} Node;
```