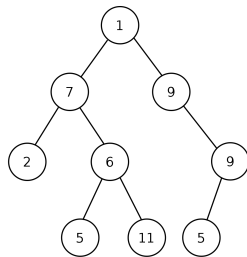


Laboratorio Fondamenti di Programmazione 2

17/11/2023

Esercizio 1

Un albero può essere rappresentato graficamente in modo testuale. Ad esempio, il seguente albero binario (non differisce molto per alberi generici):



avrà la seguente rappresentazione testuale:

```
1
 7
  2
   6
    5
    11
  9
   9
    5
```

Scrivere una funzione con segnatura:

```
template<typename T>
void stampa_albero(const AlberoB<T>&)
```

che stampi su standard output la rappresentazione grafica dell'albero in input.

Esercizio 2

Scrivere una funzione con segnatura:

```
bool pari_dispari(const AlberoB<int>&);
```

che restituisca `true` se e solo se l'albero binario soddisfa le seguenti condizioni:

- sui livelli dispari compaiono solo numeri dispari;
- sui livelli pari compaiono solo numeri pari.

Il *livello* di un nodo è definito induttivamente come:

- la radice di un albero appartiene al livello 1;
- dato un nodo appartenente al livello L , i suoi figli appartengono al livello $L + 1$.

Esercizio 3

Sia T un albero binario di interi positivi. Ogni percorso radice-foglia può essere interpretato come l'intero che si ottiene concatenando tutte le cifre nell'ordine in cui si incontrano.

Ad esempio il seguente albero binario:

```
1
 2
 3
  4
```

codifica gli interi 12 (1 -> 2) e 134 (1 -> 3 -> 4) la cui somma è 146.

Scrivere una funzione con segnatura:

```
int somma_path(const AlberoB<int>&);
```

che restituisce la somma degli interi rappresentati da ogni percorso radice-foglia di T .

Esercizio 4

Sia T un albero binario di booleani. Scrivere una funzione con segnatura:

```
bool path_congiunzione(const AlberoB<bool>& T);
```

che restituisca `true` se e solo se esiste almeno una foglia x di T tale che *tutti i nodi* nel percorso dalla radice di T a x abbiano come valore informativo `true`.

e una funzione:

```
bool path_disgiunzione(const AlberoB<bool>& T);
```

che restituisca `true` se e solo se per ogni foglia x di T esiste *almeno un nodo* nel percorso dalla radice di T a x che abbia come valore informativo `true`.