

Esercitazione - Alberi

Fondamenti 2 - Corso di Laurea in Informatica

Facendo uso della classe `AlberoB` vista a lezione, scrivere un `main C++` che implementi i seguenti metodi:

- **bool isHeap (const AlberoB<int>& a):** restituisce *true* se e solo se *a* rappresenta un heap avente il valore massimo come radice.
In particolare, per ogni nodo di *a* il valore informativo deve essere maggiore o uguale di quello di entrambi i figli;
- **bool ogniPercorsoRadiceFoglia (const AlberoB<int>& a, int max):** restituisce *true* se e solo se la somma lungo ogni percorso radice-foglia non eccede il valore massimo *max*;
- **bool pariEDispari (const AlberoB<int>& a):** restituisce *true* se e solo se sui livelli pari compaiono solo numeri interi pari e, viceversa, sui livelli dispari solo numeri dispari. Si supponga di partire dal livello 1;
- **bool vocaliEConsonanti (const AlberoB<char>& a):** supponendo che ogni percorso radice-foglia rappresenti una parola, il metodo restituisce *true* se e solo se ogni parola è ben formata. Una parola si dice ben formata se il numero di vocali e consonanti al suo interno differisce al più di uno. Si supponga che l'albero contenga solo lettere e che queste siano tutte minuscole;
- **bool fogliePosEqfoglieNeg(const AlberoB<int>& a):** restituisce *true* se e solo se il numero di foglie con valore informativo < 0 è pari al numero di foglie con valore informativo ≥ 0 ;
- **int sommaPath (const AlberoB<int>& a):** supponendo che ogni percorso radice-foglia rappresenti un numero intero, il metodo restituisce la somma di tutti i numeri rappresentati nell'albero;



PER CASA:

- **bool sommaLivelliAdiacenti (const AlberoB<int>& a, int max):** restituisce *true* se e solo se la somma per ogni coppia di livelli adiacenti non eccede il valore massimo *max*;
- **AlberoB<int> vecToAlbero (vector<int> vettore):** ricevuto in input un vector di interi, restituisce il corrispondente albero binario. Si supponga che l'albero sia completo e, dunque, che ogni nodo abbia esattamente due figli.

```
vettore = {6,1,3,4,5,2,7}
```

```
      6
     / \
    1   3
   / \ / \
  4  5 2  7
```