

# BST

## Esercizio 1:

Scrivere una funzione membro di un BST per ricercare e stampare tutti i nodi con la stessa chiave (suggerimento: quando si trova un nodo con la chiave cercata si continua a discendere nel sotto-albero di destra per cercare eventuali altri elementi con la stessa chiave).

## Esercizio 2:

Scrivere una classe Key che contenga un campo chiave di tipo int e costruire un BST con chiavi di tipo Key.

## Esercizio 3:

Scrivere una funzione membro di un BST che accetti in input una chiave e stampi la chiave del nodo predecessore e del nodo successore (se esistono) secondo un ordinamento in-order.

## Esercizio 4:

Scrivere una funzione membro di un BST che inserisca in una lista singolarmente concatenata solo gli item dei nodi di livello pari.

## Esercizio 5:

Scrivere una funzione membro di un BST per stampare il valore della chiave dell'elemento nell'albero più vicino al valore medio delle chiavi.

## Esercizio 6:

In informatica per *Look-Up Table (LUT)* si intende una struttura dati usata per sostituire operazioni di calcolo a tempo di esecuzione (runtime) con una più semplice operazione di consultazione.

Un esempio classico sono le funzioni trigonometriche: calcolare il coseno di un valore ogni qual volta serve può rallentare i processi di calcolo in certe applicazioni. Per evitare ciò l'applicazione per calcolare il valore del coseno di un valore usa la *lookup table* anziché calcolarlo usando una formula matematica. Le *LUT* sono generalmente organizzate in tabelle, ma quando contengono un numero elevato di dati conviene utilizzare degli alberi binari di ricerca, per ridurre il tempo di recupero dei dati.

Il programma da realizzare dovrà:

1. leggere dal file LUT\_cos.txt i dati di una *lookup table* della funzione coseno. I dati sono organizzati in due colonne (chiave, valore)
2. organizzarli gerarchicamente in un albero binario di ricerca (**BST**), e **bilanciarlo**
3. **stampare** i nodi del BST, visitandoli secondo la logica **inorder**
4. leggere da tastiera una chiave e stampare il relativo dato contenuto nel BST

5. se la chiave richiesta non è presente nella LUT occorre:

5.1. trovare la chiave successiva

5.2. trovare la chiave precedente

5.3. calcolare un valore interpolato tramite la formula:

$$f(x) = f(x_{prev}) + \frac{x - x_{prev}}{x_{next} - x_{prev}} (f(x_{next}) - f(x_{prev}))$$