Algoritmi e Strutture Dati (Corso di Laurea in Informatica)
Esame del 25/01/2021

UNIVERSITÀ DELLA CALABRIA

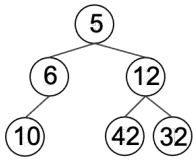
Esercizio 1

Scrivere una funzione **esercizio1** che prenda in input un albero binario di interi A e un intero x, e restituisca in output il cammino dal nodo radice dell'albero binario A al nodo foglia contenente il valore informativo x. Se non esiste un nodo foglia contenente il valore informativo x, la funzione dovrebbe restituire in output -1.

L'albero è rappresentato da una classe AlberoB. Sia a una istanza della classe, l'interfaccia è la seguente:

- `a.figlio(SIN/DES)` restituisce il figlio sinistro o destro dell'albero a se esiste, altrimenti restituisce la costante `NULLO`;
- `a.padre()` restituisce il padre dell'albero a se esiste, altrimenti restituisce la costante `NULLO`;
- `a.radice()` restituisce il valore informativo associato all'albero a (il valore intero contenuto all'interno del nodo);
- `a.foglia()` restituisce `true` se l'albero a è una foglia, altrimenti `false`.
- `a.nullo()` restituisce `true` se l'albero a è nullo, altrimenti `false`.

Si può assumere che nell'albero non vi siano due nodi con lo stesso valore informativo.

Esempio: dati in input l'albero qui d'esempio e x = 42, la funzione dovrebbe restituire il cammino 5, 12, 42. Se avessimo x = 37, la funzione dovrebbe restituire -1 poiché non esiste un nodo contenente il valore informativo 37. Se avessimo x = 12, la funzione dovrebbe restituire -1 poiché esiste un nodo con il valore informativo 12 ma esso non è un nodo foglia.	

Esercizio 2

Scrivere una funzione **esercizio2** che prenda in input un insieme di stringhe A (di cardinalità n) e un insieme di triple ordinate C dove ogni tripla contiene elementi distinti di A. La funzione deve restituire `YES` se è possibile associare ad ogni elemento $a \in A$ un numero da 1 ad n, indicato con f(a), tale che le seguenti condizioni siano vere:

- non ci sono due o più elementi con lo stesso numero, quindi f(a) != f(b) per ogni $a, b \in A$, e
- per ogni tripla $(x, y, z) \in C$, è vero che $f(x) < f(y) < f(z)$ oppure $f(z) < f(y) < f(x)$.

Se non è possibile creare una associazione che renda vere queste condizioni, si restituisca `NO`.
Si può assumere che:

- A sia rappresentato come un `vector<string>` (`ArrayList<String>` per Java),
- C sia rappresentato come un `vector<Triple>` (`ArrayList<Triple>` per Java) dove Triple è una classe che ha tre campi pubblici `x`, `y`, `z` rappresentanti rispettivamente gli elementi (stringhe) della tripla,
- Non esistano due o più triple con gli elementi nello stesso ordine.

Esempio utilizzo Triple: sia `t` una istanza di `Triple` con gli elementi (`abc, def, ghi`), allora `t.x == abc`, `t.y == def`, e `t.z == ghi`.

Esempio: in questo caso la funzione restituirà `YES` poiché è possibile associare ad ogni elemento di A un numero da 1 a 5 tale che le condizioni di cui sopra siano rispettate. In particolare, supponiamo di aver associato i seguenti numeri: a → 2, b → 1, c → 3, d → 5, e → 4. La prima condizione è soddisfatta (non ci sono due elementi con lo stesso numero); la seconda condizione è anch'essa soddisfatta, poiché: • per la tupla (a, e, d) vale f(a) < f(e) < f(d), • per la tupla (b, c, d) vale f(b) < f(c) < f(d), • per la tupla (c, a, b) vale f(b) < f(a) < f(c), • per la tupla (d, e, c) vale f(c) < f(e) < f(d).	n = 5 A = { a, b, c, d, e } C = { (a, e, d), (b, c, d), (c, a, b), (d, e, c) }

UNIVERSITÀ DELLA CALABRIA