

# Algoritmi e Strutture Dati – Prova di Laboratorio

15/06/2023

## Istruzioni

Risolvere il seguente esercizio implementando un programma in un singolo file `.cpp`, completo di funzione *main*. Si presti particolare attenzione alla formattazione dell'input e dell'output, e alla complessità target indicata per ciascuna funzionalità. Nel caso la complessità target non sia specificata, si richiede che sia la migliore possibile. La lettura dell'input e la scrittura dell'output **DEVONO** essere effettuate tramite gli stream **cin** e **cout** rispettivamente. La correzione avverrà prima in maniera automatica inviando il file `.cpp` al server indicato in aula. Quest'ultimo esegue dei test confrontando l'output prodotto dalla vostra soluzione con l'output atteso. In caso la verifica abbia esito positivo sarà possibile consegnare il compito, il quale verrà valutato dai docenti in termini di complessità. Si ricorda che è possibile testare la correttezza del vostro programma in locale su un sottoinsieme dei input/output utilizzati nella seguente maniera. I file di input e output per i test sono nominati secondo lo schema: `input0.txt output0.txt input1.txt output1.txt ...`. Per effettuare le vostre prove potete utilizzare il comando del terminale per la redirectione dell'input. Ad esempio

```
./compilato < input0.txt
```

effettua il test del vostro codice sui dati contenuti nel primo file di input, assumendo che `compilato` contenga la compilazione della vostra soluzione e che si trovi nella vostra home directory. Dovete aspettarvi che l'output coincida con quello contenuto nel file `output0.txt`. Per effettuare un controllo automatico sul primo file input `input0.txt` potete eseguire la sequenza di comandi

```
./compilato < input0.txt | diff - output0.txt
```

Questa esegue la vostra soluzione e controlla le differenze fra l'output prodotto e quello corretto.

## Esercizio

Si consideri un sistema per la gestione di alberi binari di ricerca (ABR) aventi etichette intere. Una foglia  $f$  si dice *unica* se il suo nodo padre ha uno ed un solo nodo figlio (ovvero  $f$  stesso). Dati due interi  $A$  e  $B$ , un nodo  $x$  dell'albero si dice *valido* se il sottoalbero radicato in esso contiene un numero di foglie *uniche* compreso tra  $A$  e  $B$ , estremi inclusi. Si scriva un programma che

- legga da tastiera  $N$  etichette e le inserisca all'interno dell'ABR. I valori devono essere inseriti nello stesso ordine con cui vengono letti (le etichette  $\leq$  vanno inserite a sinistra);
- stampi le etichette dei nodi *validi* ordinati per etichetta non decrescente.

L'**input** é formattato nel seguente modo: la prima riga contiene gli interi  $N$ ,  $A$  e  $B$  separati da uno spazio. Seguono  $N$  righe contenenti un'etichetta ciascuna.

L'**output** contiene gli elementi della soluzione, uno per riga.

## Esempio

### Input

```
9 1 2
10
5
3
1
7
4
8
15
20
```

### Output

```
5
7
10
15
```

