

Progetto “Albero Ternario di Ricerca”

Simulazione del 18 Maggio 2020

Descrizione del progetto

Lo scopo del progetto è l'implementazione di un albero di ricerca ternario. Questa variante di albero di ricerca permette di memorizzare una o due chiavi per ogni nodo, al contrario degli alberi binari nei quali ogni nodo contiene esattamente una singola chiave.

In un albero ternario ogni nodo contiene tre reference ad altri nodi, che indicheremo con **left** (sinistra), **center** (centro) e **right** (destra), e fino a due chiavi, che indicheremo con k_1 e k_2 , con $k_1 \leq k_2$. Un albero ternario di ricerca rispetta le seguenti proprietà:

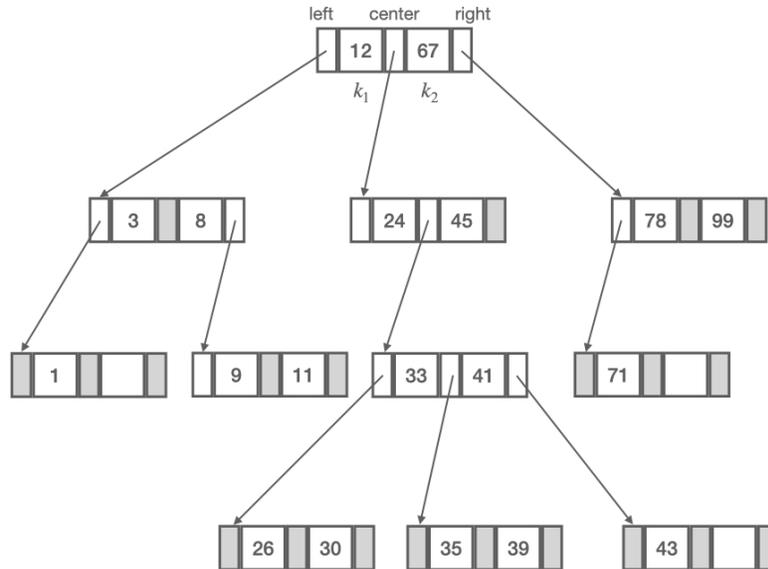
- Ogni nodo che contiene due chiavi k_1 e k_2 con $k_1 \leq k_2$ ha la proprietà che:
 - tutte le chiavi contenute nel sottoalbero di sinistra (reference **left**) sono minori di k_1 (e quindi anche a k_2)
 - tutte le chiavi contenute nel sottoalbero di centro (reference **center**) sono comprese tra k_1 e k_2 (incluso il caso in cui le chiavi siano uguali a k_1 o k_2).
 - tutte le chiavi contenute nel sottoalbero di destra (reference **right**) sono maggiori di k_2 (e quindi anche di k_1)
- I nodi contenenti una sola chiave sono necessariamente nodi foglia (ma non tutte le foglie devono contenere necessariamente una sola chiave). Questo comporta che tutti i nodi interni contengano due chiavi.

Osservando la definizione, è possibile notare che in un albero ternario di ricerca invece di doversi muovere a sinistra o destra nella ricerca (nel caso la chiave non sia già presente nel nodo corrente) diventa necessario distinguere tre casi:

- Movimento a sinistra: la chiave cercata è minore di k_1 e k_2
- Movimento centrale: la chiave cercata è compresa tra k_1 e k_2
- Movimento a destra: la chiave cercata è maggiore di k_1 e k_2

L'inserimento di una nuova chiave dovrà tenere conto del fatto che, come per la ricerca, è necessario scegliere la posizione di inserimento discendendo di volta in volta l'albero andando a sinistra, a destra o al centro. Inoltre, una volta raggiunto un nodo foglia contenente una sola chiave non è necessario creare un nuovo nodo, ma è sempre possibile inserire la chiave direttamente nel nodo.

Un esempio di albero ternario di ricerca è fornito nella seguente figura:



I campi in grigio indicano reference a `None`.

Codice

Per il progetto deve essere implementata una classe `AlberoTernario` che dovrà:

- avere un costruttore per la creazione di un albero ternario di ricerca vuoto
- consentire di inserire una nuova chiave nell'albero ternario in modo che continui a rispettare le proprietà dell'albero ternario di ricerca
- consentire di cercare una chiave nell'albero ternario, ritornando un riferimento al nodo contenente la chiave se essa esiste o `None` se la chiave non è presente nell'albero
- possedere un metodo per la stampa a schermo delle chiavi contenute nell'albero ternario dalla chiave più piccola a quella più grande (si raccomanda di osservare come sia possibile adattare la visita in ordine degli alberi binari al caso ternario)

Uno scheletro della classe da scrivere è il seguente:

```
class AlberoTernario:

    def __init__(self):
        # implementazione del costruttore

    def inserisci(self, chiave):
        # inserimento di una chiave preservando le
        # proprietà degli alberi ternari di ricerca

    def cerca(self, chiave):
        # ritorna un riferimento al nodo contenente
        # la chiave cercata (se esiste) e None altrimenti

    def stampa_in_ordine(self):
        # stampa tutte le chiavi contenute nell'albero
        # dalla più piccola alla più grande
```

Nel progetto è consentito avere funzioni aggiuntive che testano il buon funzionamento dell'albero di ricerca ternario. Non è invece necessaria l'implementazione delle operazioni di cancellazione, successore, predecessore, minimo e massimo.

Indicazioni

Il progetto deve essere svolto **individualmente**. La consegna dovrà avvenire entro le ore 23:59 del giorno xx/yy/zzzz secondo le seguenti modalità:

- Invio di una email a lmanzoni@units.it dal vostro account email istituzionale con oggetto *[informatica] consegna progetto appello del xx/yy/zzzz*.
- L'email deve avere come allegato il progetto in un singolo file in codice sorgente Python versione 3, dal nome `Nome_Cognome_matricola.py`, quindi, per esempio Mario Rossi di matricola 12345 consegnerà un file dal nome `Mario_Rossi_12345.py`.
- Il file deve contenere sotto forma di commento le seguenti linee indicanti nome, cognome e numero di matricola:

```
# Nome: Mario
# Cognome: Rossi
# Matricola: 12345
```

Note aggiuntive

Sebbene gli alberi ternari di ricerca non siano diffusi come gli alberi binari di ricerca, essi hanno alcune applicazioni e l'idea di generalizzare gli alberi binari di ricerca in modo che ogni nodo possa contenere più chiavi porta ad altre strutture dati come gli alberi 2-3-4 e i B-tree.