Indici ad Albero o (ordinati)

Lorenzo Vaccarecci

22 Marzo 2024

1 Introduzione

Organizzazione primaria : l'insieme dei file dei dati

Organizzazione secondaria : l'insieme dei file per indici ordinati

Le coppie (k_i, r_i) vengono memorizzate in un file su disco (**organizzazione secondaria**), ordinate rispetto ai valori della chiave k_i. Se l'indice è molto piccolo, può essere tenuto in memoria principale. Altrimenti è necessario tenerlo su disco, quindi per rendere più efficiente l'accesso alle coppie dell'indice si usa una struttura multilivello.
L'indice assume una **struttura ad albero** in cui ogni nodo dell'albero corrisponde a un blocco dati.

2 Requisiti

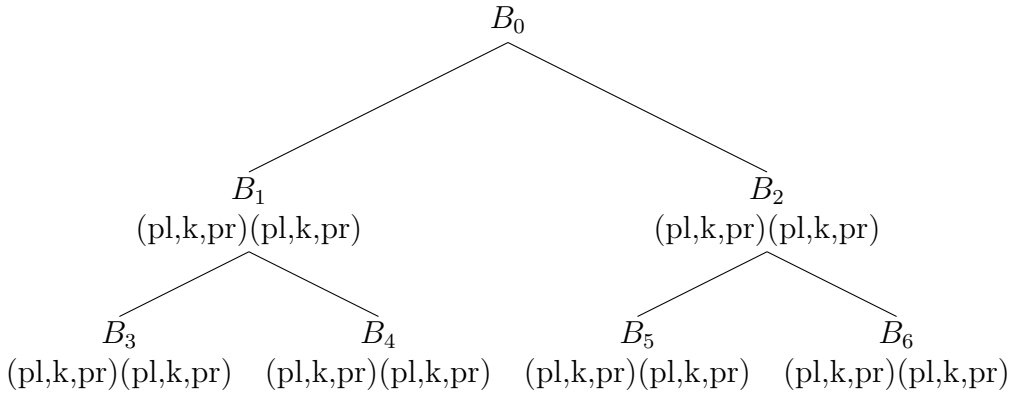
I requisiti fondamentali per indici di memoria secondaria

- **Bilanciamento**: il cammino dalla radice a ogni blocco deve essere distante uguale;
- **Occupazione minima**: ogni blocco conterrà sempre un certo numero di informazioni (pieno almeno la metà, non vogliamo blocchi vuoti);
- **Efficienza di aggiornamento**: le operazioni di aggiornamento devono avere costo limitato.

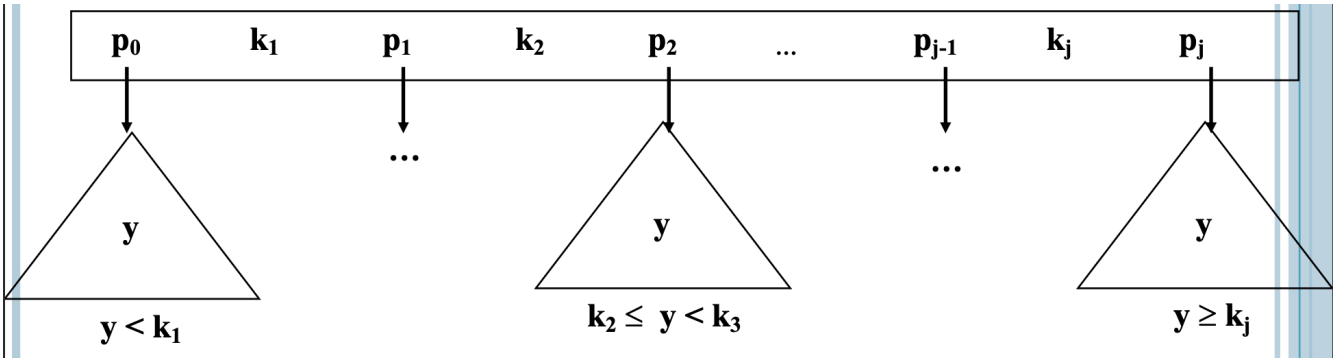
3 B$^+$-Tree

3.1 Caratteristiche

- Ogni nodo corrisponde a un blocco
- Albero bilanciato (le operazioni di ricerca, inserimento e cancellazione hanno costo, nel caso peggiore)
  - lineare nell'altezza dell'albero h
  - logaritmico: $h = O(\log n)$
- il numero massimo di elementi(=coppie) memorizzabili in un nodo è $m - 1$ dove m mi dice quanti puntatori al massimo sono contenuti in un nodo (m può essere maggiore di 2 quindi l'albero non è solo binario).
- Garantiscono che ogni blocco (tranne la radice) è pieno almeno al 50% ($\lceil \frac{m}{2} \rceil - 1$).



Nelle parentesi ci sono i puntatori ai figli. Nel caso delle foglie i puntatori punteranno all'organizzazione primaria. I nodi foglia hanno anche i puntatori ai nodi foglia successivi come una lista. I nodi puntano ai figli in base al valore delle chiavi.

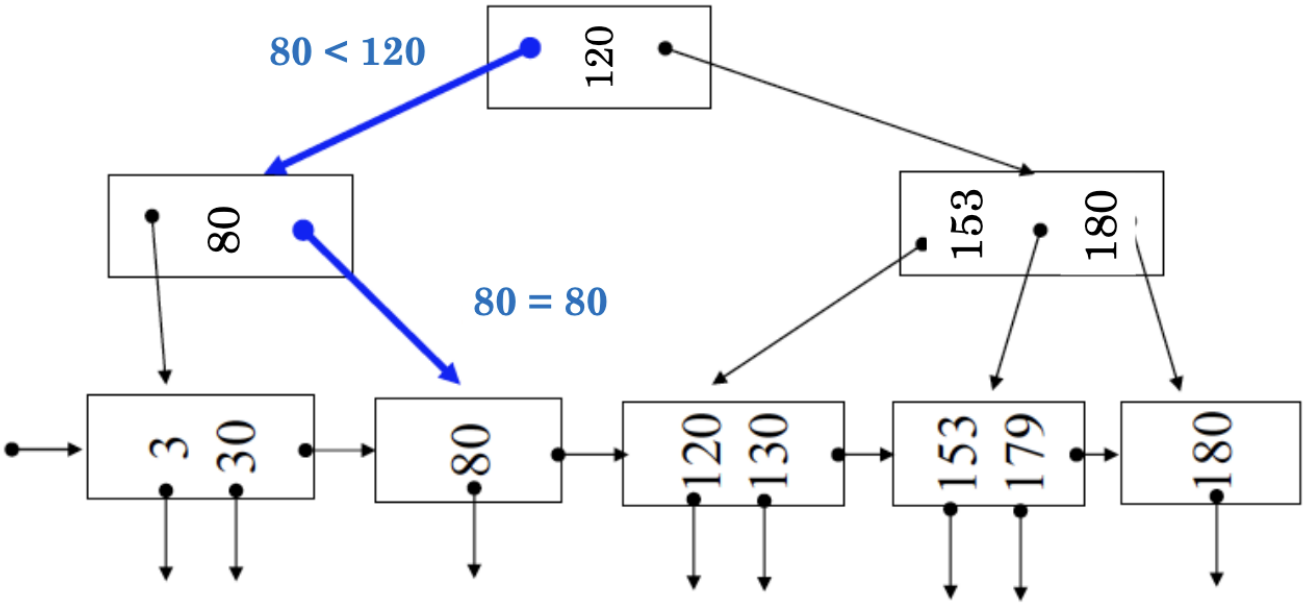


Il sottoalbero di sinistra di un separatore(k) contiene valori di chiave minori del separatore, quello destro maggiori o uguali

3.2 Ricerca

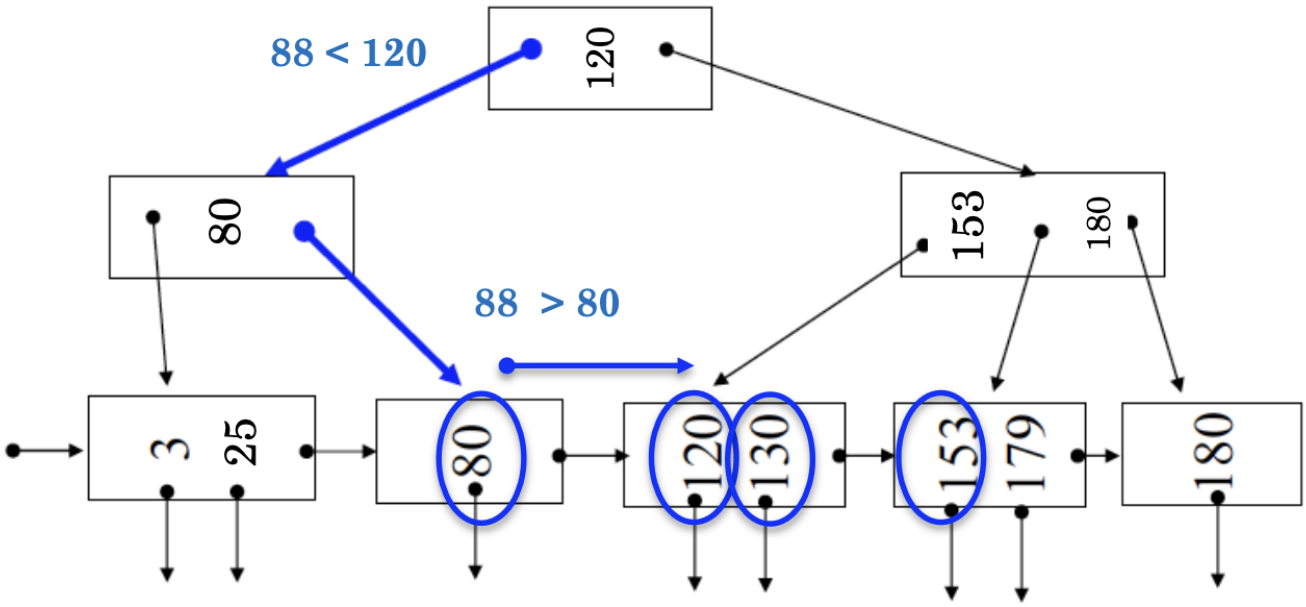
Le ricerche possono essere effettuate per indice primario e secondario, ovviamente le foglie punteranno a strutture diverse in base al tipo di indice usato.

3.2.1 Per uguaglianza



1. La radice viene caricata nel buffer e il valore viene confrontato con il valore di ricerca (K) se è minore si scende a sinistra, altrimenti centrale o destra
2. Si scende nel sottoalbero e viene controllato che il valore sia quello ricercato, se lo è scendo nel sottoalbero puntato dal numero ricercato.
3. Ritorno al passo 1 fino a quando il nodo non punta all'organizzazione primaria. Se il valore non esiste non si accede ai file dati.

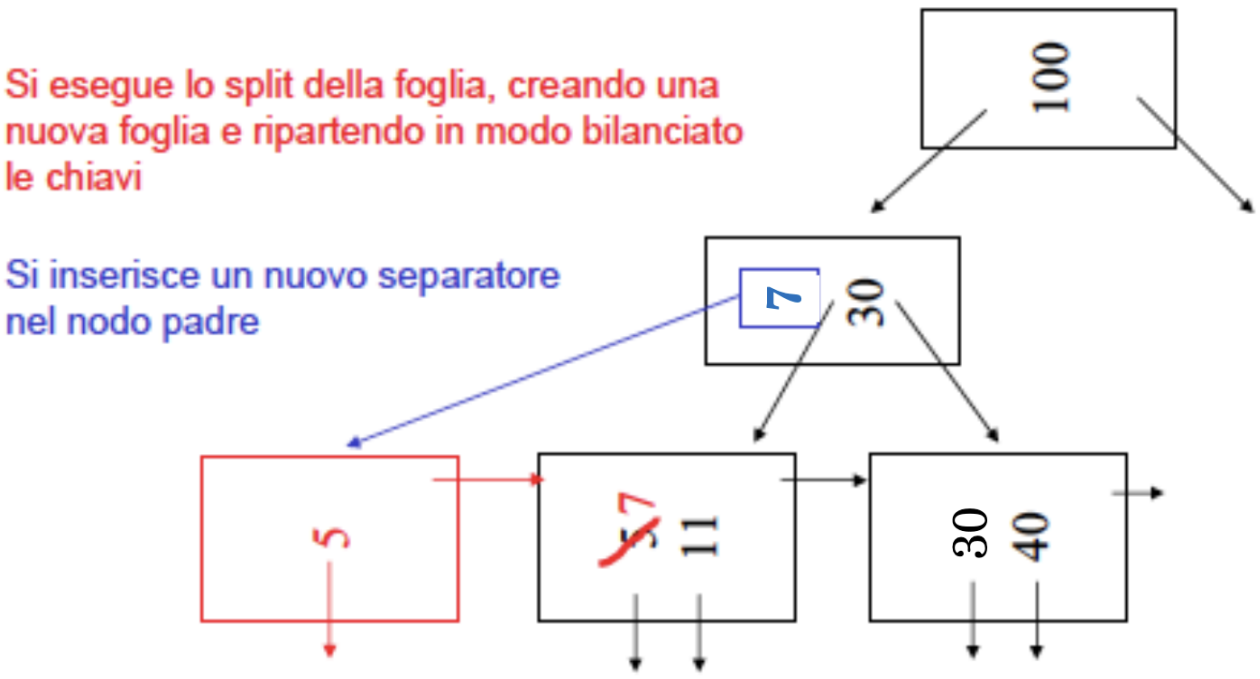
3.2.2 Per intervallo



Molto simile alla ricerca per uguaglianza solo che se non c'è controllo qual è la prima foglia a soddisfare la condizione (ad esempio nell'immagine il numero 88 è maggiore del numero più "vicino" quindi so che tutti i nodi a destra di quel nodo potrebbero soddisfare la condizione).

4 Inserimento

- Si inserisce la chiave 7



Lo split avviene nel primo nodo pieno che si incontra. Per nodo pieno si intende che non c'è spazio per inserire un nuovo separatore.

5 Cancellazione

Simile all'inserimento.