

B-tree - caratteristiche ed esercizi

Introduction

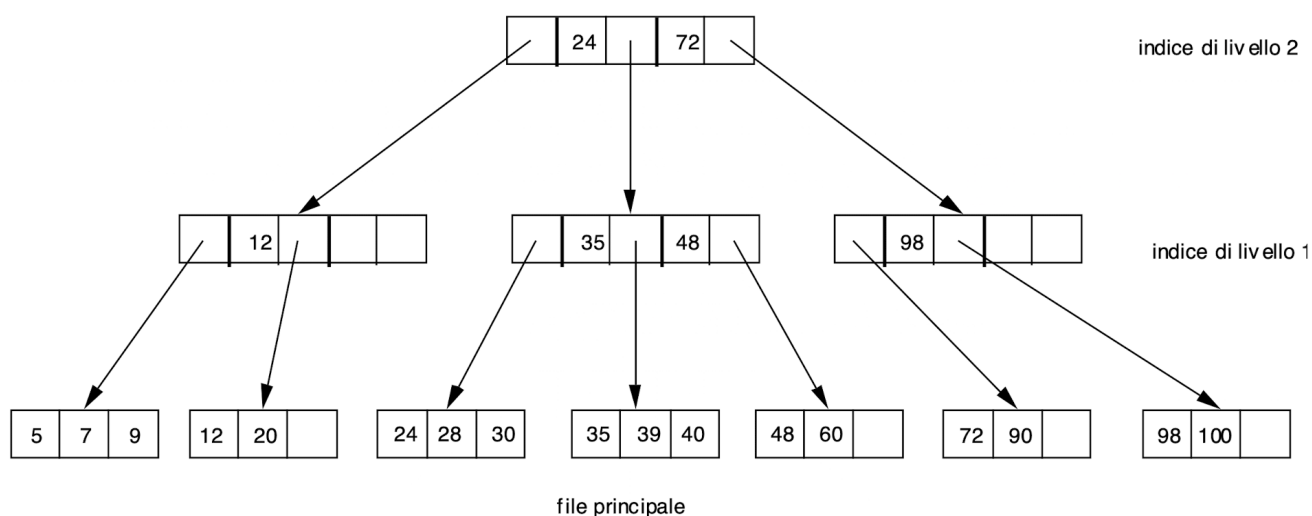
Il fatto di avere un ordinamento delle chiavi ci ha dato buoni risultati ed ha migliorato le prestazioni in termini di numero di accessi a memoria

Il **b-tree** nasce dalla **generalizzazione della struttura di indice** (estensione del file ISAM). Si accede al file attraverso una gerarchia di indici: l'indice al livello più alto nella gerarchia (la **radice**) è costituito da un **unico blocco** e quindi può risiedere in memoria principale durante l'utilizzo del file

Ogni blocco di un file indice è costituito di record contenenti una coppia (v, b) dove v è il valore della chiave del primo record della porzione del file principale che è accessibile attraverso il puntatore b ; b può essere un puntatore ad un blocco del file indice a livello immediatamente più basso oppure, nel caso sia il livello più basso del file indice, ad un blocco principale

Il primo record indice di ogni blocco contiene solo un puntatore ad un blocco le cui chiavi sono minori del blocco puntato dal secondo record indice.

Un blocco del file indice è memorizzato come quello di un ISAM; ma ogni blocco di un b-tree (indice o file principale) deve essere **pieno almeno per metà** della sua dimensione tranne eventualmente la radice (basta che contenga almeno due entrate)

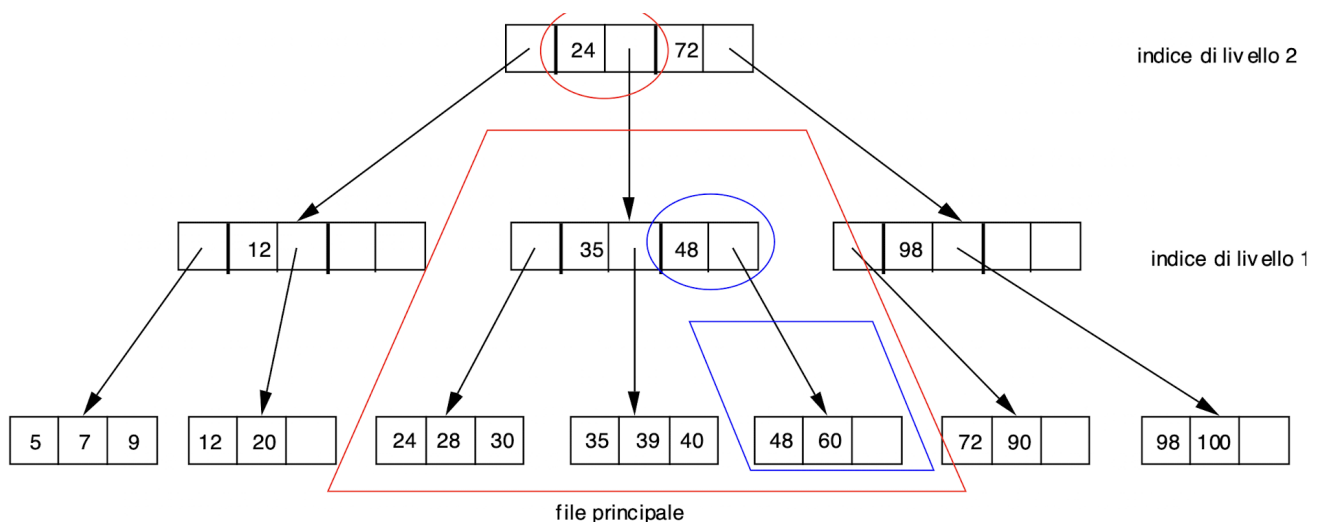


Ogni record indice ha una chiave che ricopre quelle del sottoalbero che parte dal blocco puntato

Ricerca

Durante la ricerca di un record con un dato valore per la chiave si accede agli indici a partire da quello a livello più alto; a mano a mano che si scende nella gerarchia di indice si restringe la porzione (insieme di blocchi) del file principale in cui deve trovarsi il record desiderato, fino a che, nell'ultimo livello (più basso nella gerarchia) tale porzione è ristretta ad un unico blocco

Più in dettaglio, per ricercare il record del file principale con un dato valore v per la chiave si procede nel modo seguente. Si parte dall'indice a livello più alto (che è costituito da un unico blocco) e ad ogni passo si esamina un unico blocco. Se il blocco esaminato è un blocco del file principale, tale blocco è quello in cui deve trovarsi il record desiderato; se, invece, è un blocco di un file indice, si cerca in tale blocco un valore della chiave che ricopre v e si segue il puntatore associato (che sarà o un puntatore ad un blocco dell'indice al livello immediatamente inferiore o un blocco del file principale)



Per la ricerca sono necessari $h + 1$ accessi dove h è l'altezza dell'albero (basta visitare un blocco per ogni livello)

Osservazione

L'altezza dell'albero dipende da quanto sono pieni i blocchi (dalla loro densità). Infatti più sono i blocchi più h è piccolo (e quindi meno costa la ricerca)

Come conseguenza di ciò si richiede che ogni blocco (sia del file principale sia dell'indice) sia pieno almeno per metà

Se i blocchi sono completamente pieni un inserimento può richiedere una modifica dell'indice ad ogni livello e in ultima ipotesi può far crescere l'altezza dell'albero di un livello

Massimo valore di h

Siano:

- N numero di record del file principale
- $2e - 1$ numero massimo di record del file principale che possono essere memorizzati in un blocco (e numero minimo di record per ogni foglia)
- $2d - 1$ numero massimo di record del file indice che possono essere memorizzati in un blocco (d numero minimo di record per ogni blocco indice)

Info

L'assunzione che il numero di record del file principale e del file indice che possono essere memorizzati in un blocco sia dispari viene fatta esclusivamente per rendere semplici i calcoli

L'altezza massima dell'albero denotata con k si ha quando i blocchi sono pieni al minimo, cioè quando:

- ogni blocco del file principale contiene esattamente e record
 - ogni blocco del file indice contiene esattamente d record
- Pertanto:
- il file principale ha al più $\frac{N}{e}$ blocchi
 - al livello 1 il file indice ha $\frac{N}{e}$ record che possono essere memorizzati in $\frac{N}{ed}$ blocchi
 - al livello 2 il file indice ha $\frac{N}{ed}$ record che possono essere memorizzati in $\frac{N}{ed^2}$ blocchi
 - ...
 - al livello i il file indice ha $\frac{N}{ed^{i-1}}$ record che possono essere memorizzati in $\frac{N}{ed^i}$ blocchi

Al livello k il file indice ha esattamente 1 blocco quindi $\lceil \frac{N}{ed^k} \rceil = 1$. Consideriamo per semplicità che $\frac{N}{ed^k} = 1$ da cui $d^k = \frac{N}{e}$ e infine $k = \log_d \left(\frac{N}{e} \right)$

Quindi il valore che rappresenta il limite superiore per l'altezza dell'albero è

$$k = \log_d \left(\frac{N}{e} \right)$$

Inserimento

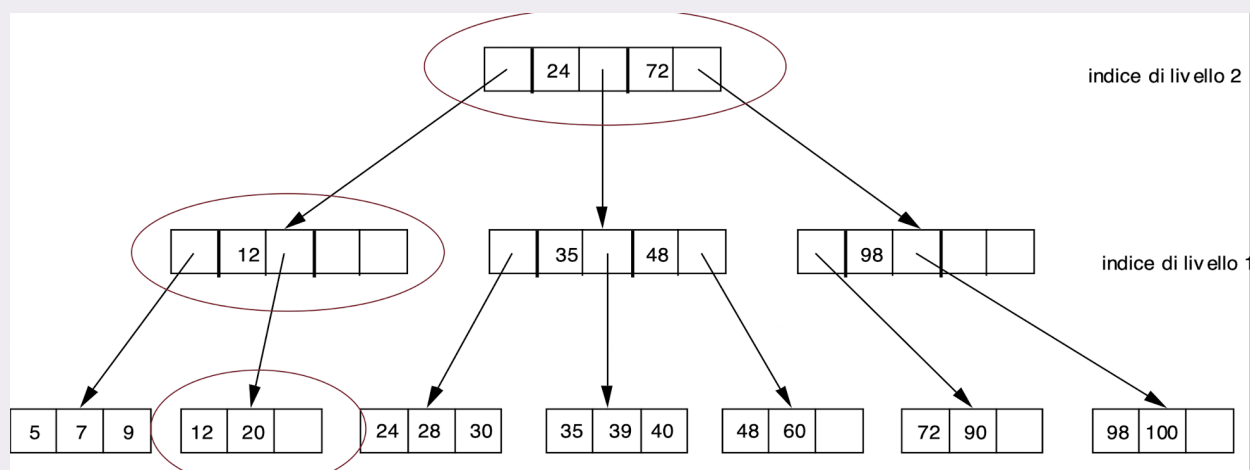
Il costo di un inserimento **se nel blocco c'è spazio sufficiente** per inserire il record è il costo di una ricerca per trovare il blocco in cui deve essere inserito il record più un accesso per riscrivere il blocco $h + 1 + 1$

Il costo di un inserimento **se nel blocco non c'è spazio sufficiente** per inserire il record è il costo di una ricerca per trovare il blocco in cui deve essere inserito il record più $s \leq 2h + 1$ (nel caso peggiore per ogni livello dobbiamo sdoppiare un blocco quindi effettuare due accessi più uno alla fine per la nuova radice)

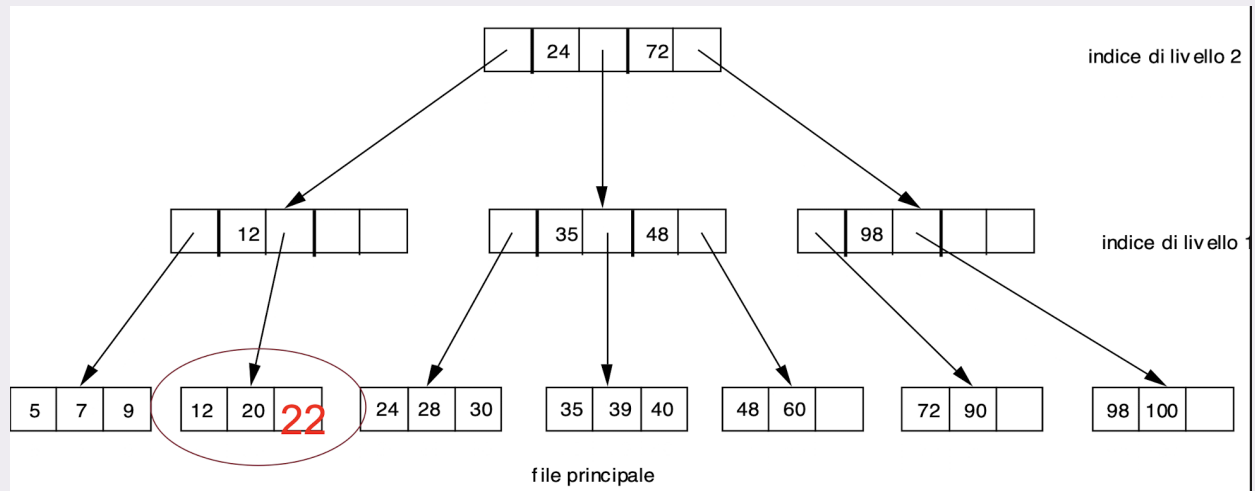
Esempi

☰ **C'è spazio sufficiente** >

Immaginiamo di voler inserire il record con chiave 22

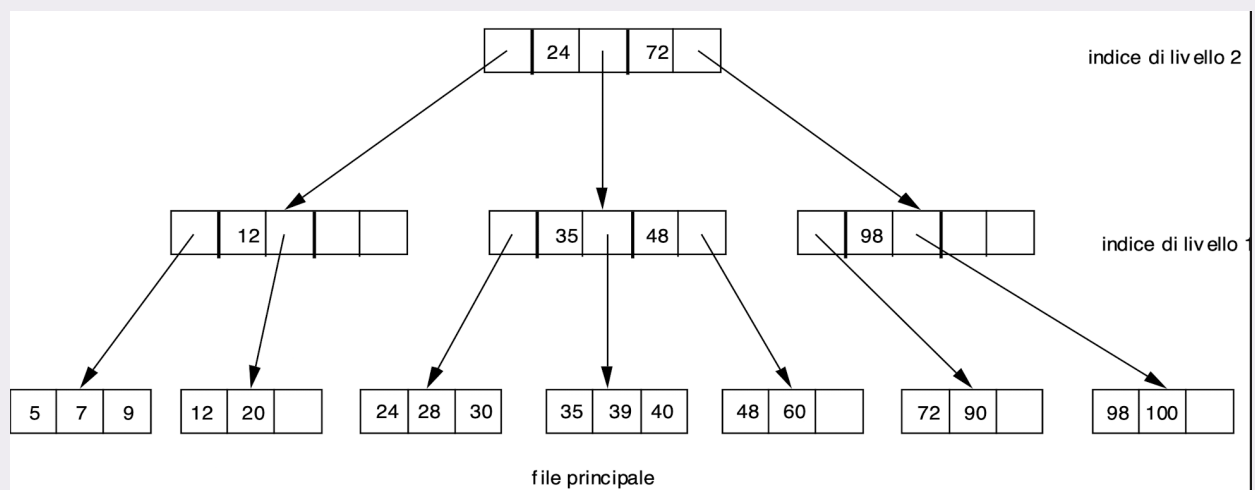


Inserendolo si ha quindi

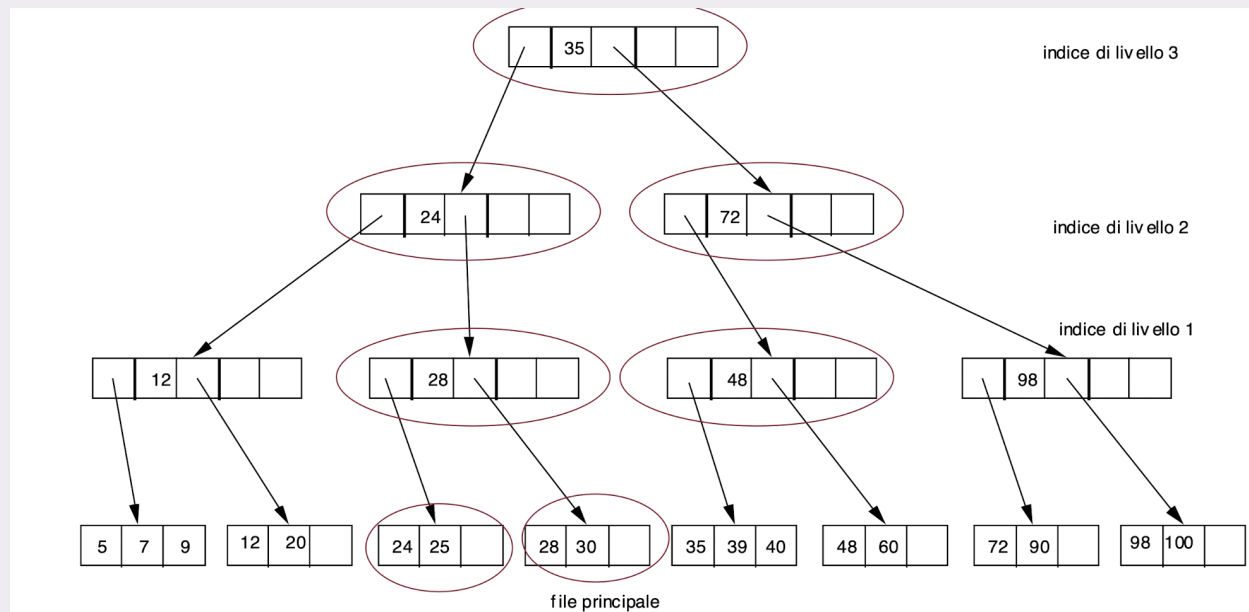


≡ Non c'è spazio sufficiente >

Immaginiamo di voler inserire il record con chiave 25



Inserendolo si ha quindi



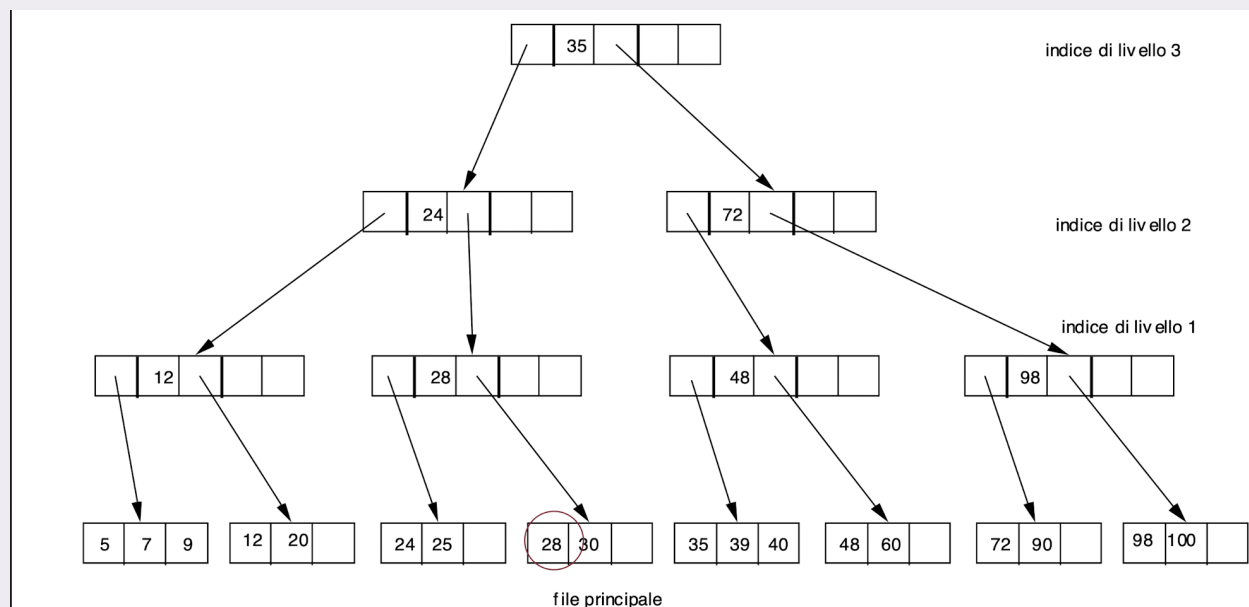
Cancellazione

Se dopo la cancellazione il blocco rimane pieno almeno per metà si ha il costo di ricerca $h + 1$ e un accesso per riscrivere il blocco $+1$, altrimenti sono necessari ulteriori accessi

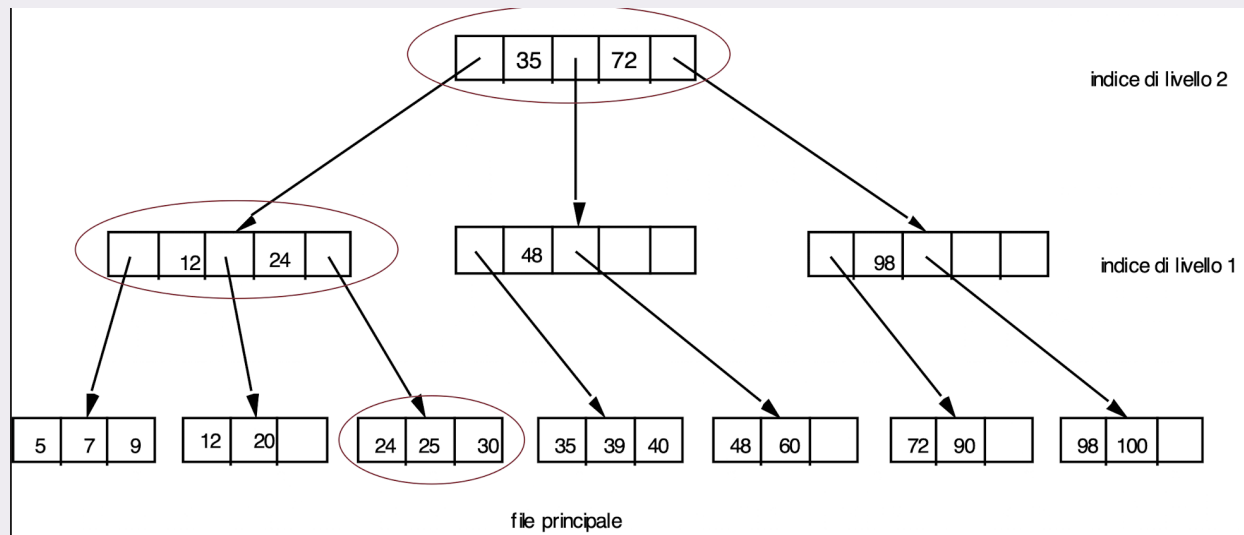
Esempi

≡ Il blocco non rimane pieno almeno per metà >

Immaginiamo di voler cancellare il record con chiave 28



Dopo la cancellazione



Modifica

Se non coinvolge campi della chiave $h + 1$ (costo della ricerca del blocco da modificare) + 1 (accesso per riscrivere il blocco), altrimenti è il costo della cancellazione più il costo dell'inserimento

Esercizi

≡ Esercizio 1 >

Supponiamo di avere un file di 170.000 record. Ogni record occupa 200 byte, di cui 20 per il campo chiave. Ogni blocco contiene 1024 byte. Un puntatore a blocco occupa 4 byte

- Se usiamo un B-tree e assumiamo che sia i blocchi indice che i blocchi del file sono pieni al minimo, quanti blocchi vengono usati per il livello foglia (file principale) e quanti per l'indice, considerando tutti i livelli foglia? Quale è il costo di una ricerca in questo caso?

Abbiamo i seguenti dati:

- il file contiene 170.000 record $\rightarrow NR = 170.000$
- ogni record occupa 200 byte $\rightarrow R = 200$
- il campo chiave occupa 20 byte $\rightarrow K = 20$

- ogni blocco contiene 1024 byte $\rightarrow CB = 1024$
- un puntatore a blocco occupa 4 byte $\rightarrow P = 4$

Calcoliamo il numero di record per blocco (file principale)

$$MR = \left\lfloor \frac{CB}{R} \right\rfloor = \left\lfloor \frac{1024}{200} \right\rfloor = 5$$

Divido il numero per 2 e prendo la parte intera per prendere il numero di record occupati

$$e = \left\lfloor \frac{5}{2} \right\rfloor = 3$$

Calcolo il numero di blocchi per il file principale (ovvero il numero di record del file indice)

$$BF = \left\lceil \frac{NR}{e} \right\rceil = \left\lceil \frac{170.000}{3} \right\rceil = 56667$$

Calcoliamo il numero di record memorizzabili in un blocco del file indice; per farlo all'inizio tolgo i 4 byte del record puntatore dalla metà della dimensione totale del blocco e alla fine lo riaggiungo per raggiungere il riempimento richiesto

$$d = \left\lceil \frac{CB/2 - P}{K + P} \right\rceil + 1 = \left\lceil \frac{512 - 4}{24} \right\rceil + 1 = 22 + 1 = 23$$

Calcoliamo il numero di blocchi per il file indice al primo livello

$$B_1 = \left\lceil \frac{BR}{d} \right\rceil = \left\lceil \frac{56667}{23} \right\rceil = 2464$$

Calcoliamo il numero di blocchi per il file indice al secondo livello

$$B_2 = \left\lceil \frac{B_1}{d} \right\rceil = \left\lceil \frac{2464}{23} \right\rceil = 108$$

Calcoliamo il numero di blocchi per il file indice al terzo livello

$$B_3 = \left\lceil \frac{B_2}{d} \right\rceil = \left\lceil \frac{108}{23} \right\rceil = 5$$

Calcoliamo il numero di blocchi per il file indice al quarto livello

$$B_4 = \left\lceil \frac{B_3}{d} \right\rceil = \left\lceil \frac{5}{23} \right\rceil = 1$$

A questo punto mi fermo poiché ho trovato la radice (livello con un solo blocco)

Complessivamente quindi si avranno

$$BI = B_1 + B_2 + B_3 + B_4 = 2464 + 108 + 5 + 1 = 2578$$

Il costo della ricerca sarà 5 (un blocco per ognuno dei 4 livelli di indice +1 blocco per il file principale)

☰ **Esercizio 2** >

Supponiamo di avere un file di 170.000 record. Ogni record occupa 200 byte, di cui 20 per il campo chiave. Ogni blocco contiene 1024 byte. Un puntatore a blocco occupa 4 byte

- Se usiamo un B-tree e assumiamo che sia i blocchi indice che i blocchi del file sono pieni al massimo, quanti blocchi vengono usati per il livello foglia (file principale) e quanti per l'indice, considerando tutti i livelli foglia? Quale è il costo di una ricerca in questo caso?

Abbiamo i seguenti dati:

- il file contiene 170.000 record $\rightarrow NR = 170.000$
- ogni record occupa 200 byte $\rightarrow R = 200$
- il campo chiave occupa 20 byte $\rightarrow K = 20$
- ogni blocco contiene 1024 byte $\rightarrow CB = 1024$
- un puntatore a blocco occupa 4 byte $\rightarrow P = 4$

Calcoliamo il numero di record per blocco (file principale)

$$MR = \left\lfloor \frac{CB}{R} \right\rfloor = \left\lfloor \frac{1024}{200} \right\rfloor = 5$$

Poiché è riempito al massimo non devo togliere alcun blocco

Calcolo il numero di blocchi per il file principale (ovvero il numero di record del file indice)

$$BF = \left\lceil \frac{NR}{e} \right\rceil = \left\lceil \frac{170.000}{5} \right\rceil = 34000$$

Calcoliamo il numero di record memorizzabili in un blocco del file indice

$$d = \left\lceil \frac{CB/2 - P}{K + P} \right\rceil + 1 = \left\lceil \frac{1024 - 4}{24} \right\rceil + 1 = 22 + 1 = 43$$

Poiché è riempito al massimo non devo togliere alcun blocco

Calcoliamo il numero di blocchi per il file indice al primo livello

$$B_1 = \left\lceil \frac{BR}{d} \right\rceil = \left\lceil \frac{34000}{43} \right\rceil = 791$$

Calcoliamo il numero di blocchi per il file indice al secondo livello

$$B_2 = \left\lceil \frac{B_1}{d} \right\rceil = \left\lceil \frac{791}{43} \right\rceil = 19$$

Calcoliamo il numero di blocchi per il file indice al terzo livello

$$B_3 = \left\lceil \frac{B_2}{d} \right\rceil = \left\lceil \frac{19}{43} \right\rceil = 1$$

A questo punto mi fermo poiché ho trovato la radice (livello con un solo blocco)

Complessivamente quindi si avranno

$$BI = B_1 + B_2 + B_3 = 791 + 19 + 1 = 2578$$

Il costo della ricerca sarà 4 (un blocco per ognuno dei 3 livelli di indice +1 blocco per il file principale)