Si consideri le relazioni R(A, B, C, D) e la seguente query

SELECT MIN(A) FROM R WHERE B=10

Quale dei seguenti indici garantisce l'efficienza massima?

Dato che si cita una WHERE, di sicuro conviene usare un indice B-Tree o B+ Tree.
L'esercizio pone come scelte due indici hash e due indici B-TREE.
La coppia che ci interessa è chiaramente (B,A) e ci interessa creare un indice partendo da B per velocizzare la ricerca (piuttosto quindi che fare un indice su A,B).
Per la scelta:

- la hash si pone valida come ricerca solo nel caso di paragoni in senso di uguaglianza (perché hash è veloce nel ricercare *esattamente* un certo valore)
- i B-Tree o B+ Tree si pongono valide nei casi in cui ci siano degli operatori di confronto/aggregazione e generalmente opero *su più valori*, buono soprattutto quando devo confrontare frequentemente dei valori (es. nelle chiavi, dove di default si ha questo tipo di indice)

Quindi:

4. Indice B-TREE sulla coppia (B,A)

Data la query `SELECT * FROM R WHERE C='Valore'` sulla relazione R(A, B, C). Quale dei seguenti indici in genere assicura le migliori performance in termini di velocità dell'esecuzione della query?

Stiamo agendo su una WHERE e si pone come una ricerca. La hash esegue la ricerca in tempo O(1).
Dunque, la risposta possibile qui è:

(1) Indice Hash su C;

Data la query `SELECT * FROM S WHERE X=4 AND Z>8` sulla relazione `S(X, Y, Z, W)`. Quale dei seguenti indici in genere assicura le migliori performance in termini di velocità dell'esecuzione della query?

1. Indice Hash su (Z,X)
2. Indice B+Tree su X
3. Indice Hash sulla coppia (X,Z)
4. Indice B+Tree sulla coppia (X,Z)

Come descritto sopra, opero su una coppia di valori che devono essere confrontati. Ciò mi porta a scegliere B+ Tree come indice di riferimento.

Data la relazione e R(A, B, C) la query SELECT * FROM R ORDER BY C, quale dei seguenti indici velocizza l'esecuzione della query?

Stiamo confrontando una serie di valori con C; pertanto, la hash qui non sarebbe tanto efficace quanto un indice B+ Tree; esso inoltre viene fatto su C, quindi la risposta è:

(2) Indice B+Tree su C