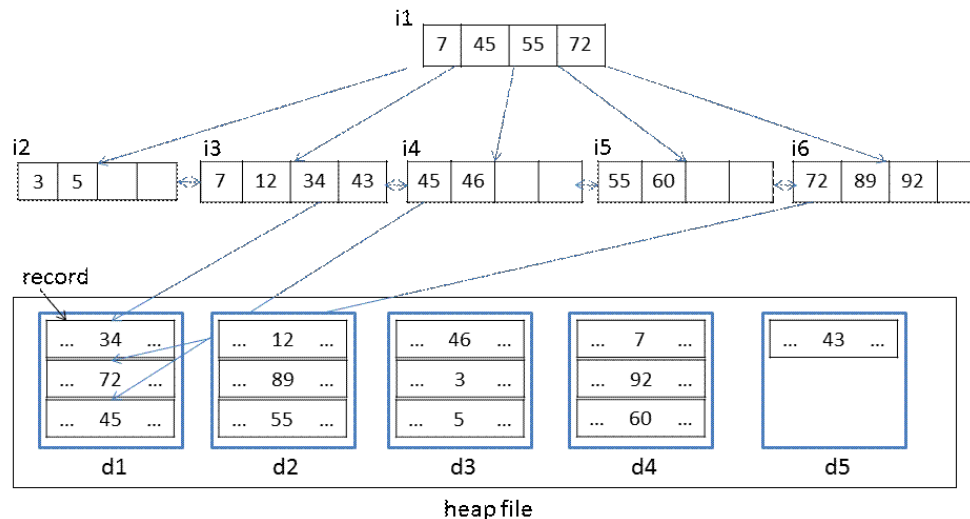


1. 以下の図は、テーブル Members(id, name, address, age)のレコードをヒープファイルに格納し、レコード中の属性 age に対して B+-tree 索引を構成している様子を示している。図中、i1~i6 は索引ページ番号、d1~d5 はデータページ番号である。



※ d2 以降、索引からレコードへのポインタは省略

- (1) 上記の B+-tree 索引は「二次索引」と呼ばれる。二次索引とはどのような索引か、上の図を用いて説明せよ。

リーフノードに値そのものが入っておらず、実際のデータの保存場所へのポインタが格納されている索引。上の図で一次索引はヒープファイルで構成されており、B+-tree のリーフノードはヒープファイル内のデータの場所へのポインタが入っている。

- (2) このテーブルに対して以下の問合せをする場合を考える。

SELECT * FROM Members WHERE age > 45 and age < 80;

この時ディスクから読み込むページ (索引, データの両方) のページ番号を求めよ。

i1, i4,i5,i6,d3,d2,d4,d1

- (3) (2)の問合せで読み込まれたデータはバッファプールに全て格納されているとする。

ここで以下の問合せが行われたときディスクから読み込むページのページ番号を求めよ。

SELECT * from Members WHERE age >=60;

なし

- (4) 下記のテーブルに対して age 属性の値が 20 であるレコードを挿入した後の B+-tree 索引を求めよ。なお、リーフでないノードの分割のアルゴリズムは以下のとおりである。

```

proc divide_nonleaf ( N )
  input
    N : 分割する non-leaf ノード
  output
    newEnt : 親ノードに挿入するエントリ
  process:
    ノード M を作る
      N の d+2 番目から 2d+1 番目までの key 値を M へ移動
      N の d+2 番目から 2d+2 番目までのポインタを M へ移動
      N から M へ兄弟ポインタを張る
    newEnt = <N の d+1 番目の key 値, M へのポインタ>
    N の d+1 番目の key 値を削除
    return newEnt;

```

2. ページ数がそれぞれ M, N である二つのテーブルを、ソートマージ結合アルゴリズムを用いて結合するとき、その IO コストは $2M \log_k M + 2N \log_k N + M + N$ である (k は k -way マージソートのパラメタ値)。この IO コストの理由をソートマージ結合アルゴリズムの処理の流れをもとに記述せよ。

(省略)

3. 3 種類のテーブル Movie, Actor, Cast が以下のレコード数、ページ数で格納されている。それぞれの物理格納方式は下図の右側に示すとおりである。

Movie (mid, title, director)

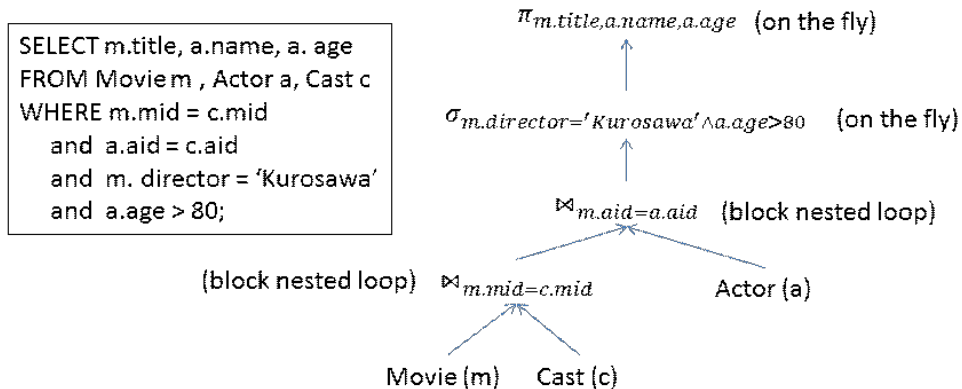
Actor (aid, name, age)

Cast (mid, aid)

	# of records	# of pages
Movie	10,000	5
Actor	50,000	25
Cast	300,000	60

- Movie
 - mid の値によるハッシュファイル
 - director 値の異なり数は 1000
- Cast
 - mid の値によるハッシュファイル
 - aid の値による二次索引
 - 1 movie の平均 actor 数は 30
- Actor
 - aid の値によるハッシュファイル
 - age の値の幅 {0, 100}
- 結合に使えるバッファ数: 12

このテーブルに対して以下の問合せを行った場合、最も素朴な問合せプランは下図の右に示すとおりである。下記のプランより効率の良いプランを、理由とともに記述せよ。なお、回答において正確なコスト計算は必ずしも必要ではなく、下記プランより問合せコストが改善される十分な理由が示されていればよい。



(省略)

4. (1) 障害時回復機能と同時実行制御機能が DBMS に欠かせない理由を ACID 特性の全ての特性名とその説明を文書内に含めて説明せよ。(省略)

- (2) 以下の二つのスケジュール S1,S2 の相反グラフをそれぞれ書き,
相反直列化可能かどうかを求めよ

$S1 = r1(A)r2(B)r3(C)w1(B)c1w2(C)c2w3(A)c3$

$S2 = r1(A)r2(B)r3(C)r1(B)r2(C)r3(D)w1(A)c1w2(B)c2w3(C)c3$

S1 は相反直列化不可能、S2 は相反直列化可能

- (3) 上記のスケジュール S1 に二相ロックを適用した場合のスケジュール S'1 を記述せよ。また S'1 の相反グラフを求め、相反直列化可能かどうかを示せ。なおデッドロックにはまった場合は wait-die 方式でデッドロックを解除すること。トランザクションの開始時間は各トランザクションの最初の操作が行われる時間とする。