Exercise (static multilevel index, Sequential file organization)

Let r=(A,B,C,D,E), with A a key.
Assume

nr = 500.000	number of records in the relation
Lr = 200 byte	size of a record (fixed length records)
LA = 16 byte	size of attribute A
Lp = 4 byte	size of a pointer
Lb = 2400 byte	size of a block

Sequential file organization on search-key A

1. Show the number of blocks of the file.

2. Show the number of blocks of a static multilevel index on search-key A (primary index).

3. Outline the steps in answering the following queries and the cost in terms of number of block transfers from disk:

1) select * from r where A=xxx;

2) select * from r where 320.000 <= A < 330.000;
assuming A uniformly distributed on the interval [200.000; 800.000]

Point 1.
We evaluate the blocking factor of the relation.

$$f_r = \left\lfloor \frac{Lb}{Lr} \right\rfloor \qquad f_r = \left\lfloor \frac{2.400}{200} \right\rfloor = 12$$

blocking factor of the relation r
max number of records in a block of the file

The number of blocks of the file is:

$$n_b = \left\lceil \frac{nr}{f_r} \right\rceil \qquad n_b = \left\lceil \frac{500.000}{12} \right\rceil = 41.667$$

number of blocks of the file

Point 2
Records are stored in search-key order. The primary index for the file is sparse.
The number of search-key values in the index is equal to the number of blocks of the file.

We evaluate the number of index records (key, point) in a block or blocking factor of the index, named f_I

$$f_I = \left\lfloor \frac{Lb}{LA + Lp} \right\rfloor \qquad f_I = \left\lfloor \frac{2.400}{20} \right\rfloor = 120$$

$$n_{blevel1} = \left\lceil \frac{41667}{120} \right\rceil = 348$$

number of blocks of the first level of the index

$$n_{blevel2} = \left\lceil \frac{348}{120} \right\rceil = 3$$ *number of blocks of the second level of the index*

$$n_{blevel3} = \left\lceil \frac{3}{120} \right\rceil = 1$$ *number of blocks of the third level of the index*

We have a three-level sparse index.
The number of blocks of the index is:

$$n_{blndex} = 348 + 3 + 1 = 352$$

Point 3.1

select * from r where A=xxx;

- Cost of the query using the index.

C = height of the index + 1 block for the file
C = 3 + 1 = 4

- Cost of the query using binary search.

$C' = \lceil \log_2 n_b \rceil = \lceil \log_2 41.667 \rceil = 16$

Cost of the query: min(C, C') = min(4, 16) = 4
The query optimizer uses the primary index.

Point 3.2

select * from r where 320.000 <= A < 330.000;

- Cost using the index

We evaluate the selectivity factor of the query
f_s = (330.000-320.000) / (800.000 -200.000) = 10.000 / 600.000 = 1/60

If blocks are chained together in search-key order:

C = number of the index levels + $\lceil fs * n_b \rceil$

Number of file block transfers:

$\lceil fs * n_b \rceil = \lceil 1/60 * 41667 \rceil = 695$

Cost of the query:
C = 3 + 695 = 698 (sequential file organization, records are stored in search-key order)

If blocks are not chained:

$C = \lceil fs * n_{blevel3} \rceil + \lceil fs * n_{blevel2} \rceil + \lceil fs * n_{blevel1} \rceil + \lceil fs * n_b \rceil$
$C = \lceil 1/60 * 1 \rceil + \lceil 1/60 * 3 \rceil + \lceil 1/60 * 348 \rceil + \lceil 1/60 * 41667 \rceil = 1 + 1 + 6 + 695 = 703$

The cost of the query is 703 block transfers.

Exercise (static multilevel index, heap file organization)
Let's suppose we have a heap file organization on search-key A.
If we build an index on search-key A, the index is dense with an entry for every search-key value in the file.
A is a key of r. As a consequence, the number of search-key values in the index is equal to the number of records in the file (500.000).

$f_I = 120$

$n_{blevel1} = \left\lceil \frac{500.000}{120} \right\rceil = 4167$ *number of blocks of the first level of the index*

$n_{blevel2} = \left\lceil \frac{4167}{120} \right\rceil = 35$ *number of blocks of the second level of the index*

$n_{blevel3} = \left\lceil \frac{35}{120} \right\rceil = 1$ *number of blocks of the third level of the index*

We have a three-level dense index.
The number of blocks of the index is:

$n_{blndex} = 4167 + 35 + 1 = 4.203$

Point 3.1

select * from r where A=xxx;

Cost of the query using the index

C = height of the index + 1 block for the file
C = 3 + 1 = 4

Point 3.2

select * from r where 320.000 <= A < 330.000;

- Cost of the query using the index
$f_s = 1/60$ selectivity factor of the query

$C = \lceil fs * n_{blevel3} \rceil + \lceil fs * n_{blevel2} \rceil + \lceil fs * n_{blevel1} \rceil + \lceil fs * nr \rceil$ *(one block transfer for each record)*
$C = \lceil 1/60 * 1 \rceil + \lceil 1/60 * 35 \rceil + \lceil 1/60 * 4167 \rceil + \lceil 1/60 * 500.000 \rceil =$
$1 + 1 + 70 + 8.334 = 8.406$

- Cost of sequential scan of the file
$C = n_b = 41.667$

Cost of the query: min(C, C') = min (8.406, 41.667) = 8.406
The query optimizer uses the secondary index.