Exercise (Hash index)

Let us suppose we have the following relation r=(A,B,C), with A the primary key.
Let's A be a sequence of upper-case letters in the alphabet
Assume

nr = 10	number of records in the relation
Lr = 200 byte	size of a record (fixed length record)
LA = 96 byte	size of attribute A
Lp = 4 byte	size of a pointer
Lb = 500 byte	size of a block

Heap file organization.

1) Choose a simple hash function on search-key A, assuming blocks of the index with fill factor 80% .

2) Assume the file is empty and the following records are stored in the file in the same order in which they are listed (the search-key value is shown):
AA…A, CC…C, EE…E, FF…F, BB…B, II…I, NN…N, QQ…Q, TT…T, DD…D.
Show the index and the pointers to the file.

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='XX…X';

2) select * from r where 'XX…X' <= A < 'YY…Y';

4) Evaluate the overflow probability.

Point 1

Number of blocks of the index

$$f_I = \left\lfloor \frac{Lb}{LA + Lp} \right\rfloor \qquad f_I = \left\lfloor \frac{500}{100} \right\rfloor = 5$$

blocking factor of the index: max number of records in a block of the index

$$f'_I = \left\lfloor \frac{5 * 80}{100} \right\rfloor = 4$$

number of records in a block of the index assuming fill factor 80%

$$n_{bI} = \left\lceil \frac{nr}{f'_I} \right\rceil \qquad n_{bI} = \left\lceil \frac{10}{4} \right\rceil = 3$$

number of blocks of the index

We have that the hash function distributes records into 3 blocks.
Let h denotes the hash function:

h: A -> {0, 1, 2}

Typical hash functions perform computation on the internal binary representation of characters in the search key.
In the following, for simplicity, we use the first character of the key.

$$h(A) = (A[0] - \text{'A'}) \,\%3$$

Point 2

Number of blocks of the file

$f_r = \left\lfloor \frac{Lb}{Lr} \right\rfloor$ $\quad f_r = \left\lfloor \frac{500}{200} \right\rfloor = 2$ *blocking factor of the relation r: max number of records in a block of the file*

$n_b = \left\lceil \frac{nr}{f_r} \right\rceil$ $\quad n_b = \left\lceil \frac{10}{2} \right\rceil = 5$ *number of blocks of the file*

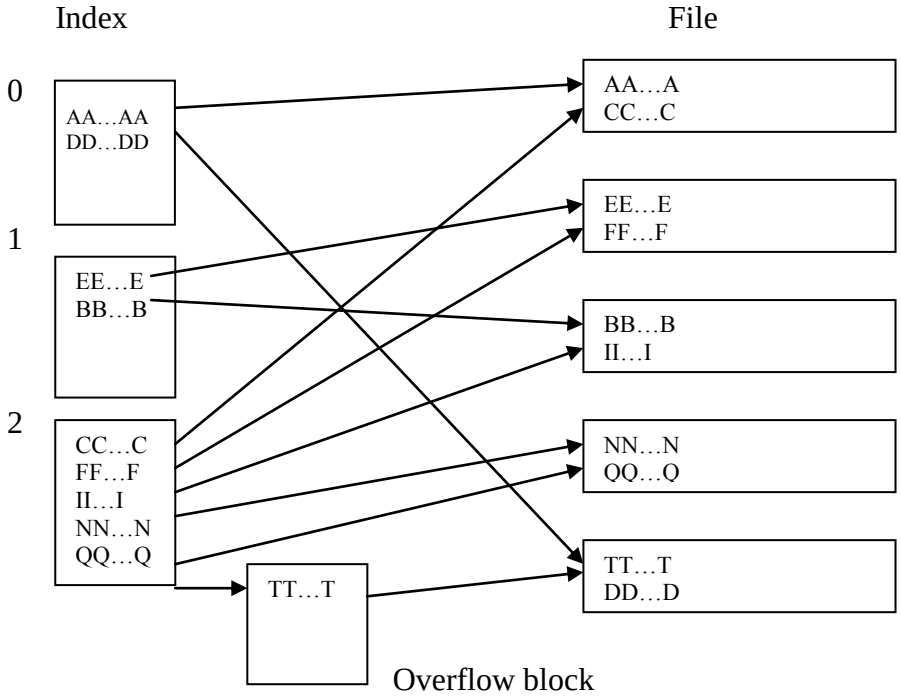
Records are stored as shown below.

Block 0	AA…A CC…C
Block 1	EE…E FF…F
Block 2	BB…B II…I
Block 3	NN…N QQ…Q
Block 4	TT…T DD…D

To construct the index, we apply the hash function.

h(AA..A)=h(DD…D)= 0
h(EE…E)=h(BB…B)=1
h(CC…C)=h(FF…F)=h(II…I)=h(NN…N)=h(QQ…Q)=h(TT…T)=2

We assume overflow is handled using separate blocks.
The index is the following:



Point 3.1

select * from r where A=”XXXXX”

We use the index.
Best case cost: C = 2 1 block of the index + 1 block of the file
Worst case cost: C = 3 1 block of the index + 1 overflow block + 1 block of the file
On average: C = ⌈(2 * 2 + 1* 3)/ 3 ⌉= 3

Point 3.2

select * from R where “XXXXX” <= A <=”YYYYY”

The index is not used.
Sequential scan of the file:

$$C = n_b = 5$$

Point 4

Hash function h: K -> {0, .., n_b-1}
Let’s us apply the hash function to a record.

$p = 1/n_b$ probability that the hash function generates block j
1-p probability that the hash function generates a block different from j

Given nr records, the probability that a block is generated x times is the probability that the hash function generates the same block for x records:

$$P(x) = \binom{nr}{x} (p)^x (1-p)^{nr-x}$$

Overflow condition: block generated more times than the blocking factor of the index (f_I)

Probability of overflow:

$$\sum_{x > fI} P(x)$$