Gradiance Online Accelerated Learning

Dian Hao

- Home Page
- Assignments Due
- Progress Report
- Handouts
- Tutorials
- Homeworks
- Lab Projects
- Log Out

Help

Submission number:	613384
Submission certificate:	FE177525
Submission time:	2022-02-05 05:06:07 PST (GMT - 8:00)

Number of questions:	4
Positive points per question:	1.0
Negative points per question:	0.0
Your score:	3

hash and sort

1. Suppose that we are maintaining a table using ***linear hashing***, with the hash function h(x) = x mod 8. Each block can store up to 3 records. The constraint on the number of buckets is R <= 2.5N, where R is the total number of records in the structure, and N is the number of buckets.

Construct the hash structure after the records with the following keys have been inserted:

(2, 3, 5, 7, 11, 17, 19, 23, 29, 31)

Construct the final structure. Be sure to indicate the value i (the number of hash bits being used), N, and R. Which of the following is true?

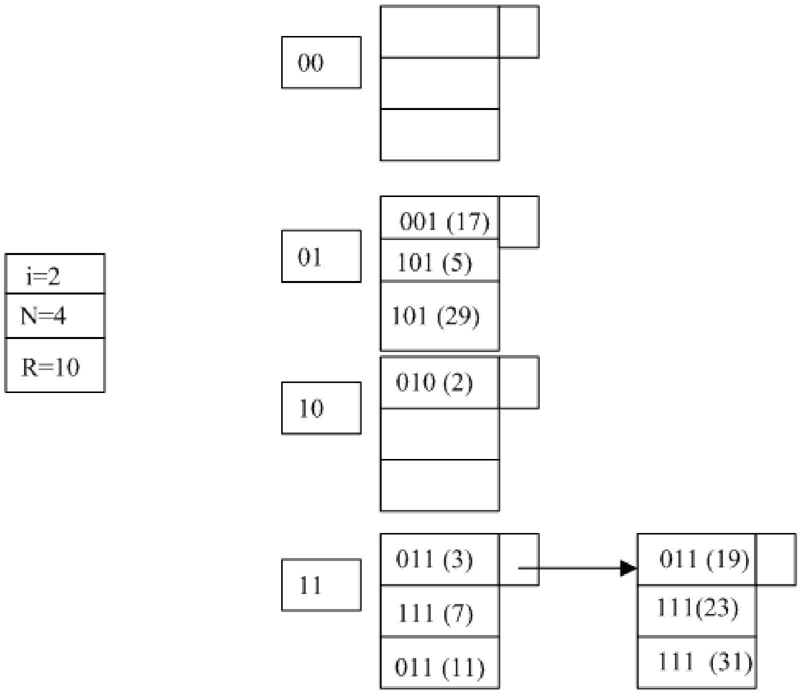
a) There is an empty bucket.
b) Key value 7 is in a block by itself.
c) Key value 11 is in a block by itself.
d) Key value 31 is in a block by itself.

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

The final hash structure is:



Therefore:

- There are 4 non-overflow blocks, 1 overflow block.
- Key values 17, 5, and 29 are in the same block.
- Key value 2 is in a block by itself.
- Key values 3, 7 and 11 are in the same non-overflow block, and key values 19, 23, and 31 are in the same overflow block.
- There is an empty bucket.

2. We can generalize two-pass multiway merge-sort to use more than two passes. On the first pass, we divide the file into as many groups as we can merge in one pass. Then, recursively sort each group, using as many passes as necessary, and finally merge the sorted groups.

A file of 500 blocks is to be externally sorted, using 40 main-memory buffers. How many passes are needed? How many runs remain at the last pass? Find the correct number in the list below.

a) The number of passes required is 2.

b) The number of runs at the last pass is 39.

c) The number of runs at the last pass is 12.

d) The number of passes required is 3.

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

The first pass divides the file into 13 runs of 40 blocks each. The degree of merging is 39, so we must repeatedly divide 13 by 39, raising to the next higher integer if the result is not an integer. When we reach a single run, no more passes are needed. In this case, the total number of passes, including the first where the runs are created, is 2, and the last pass merges 13 runs.

3. Consider a linear hash table that uses 4-bit hash keys and stores two records per bucket. The capacity threshold is 75%; that is, we create a new bucket whenever the number of records is more than 3/2 the current number of buckets. Simulate the insertion, into an initially empty hash table, of records with (hash values of) keys 1111, 1110, 1101,..., 0001, 0000, in that order. Which of the following keys **never** appears alone in a bucket at any time during the insertion sequence?

a) 0010

b) 1100

c) 0110

d) 1000

Answer submitted: **a)**

Your answer is incorrect.

You need to simulate the entire sequence of insertions and record which keys ever appear by themselves. In particular, 0010 appears by itself in bucket 0010 when 0000 is inserted. Linear hashing is covered in Sects. 14.3.7 and 14.3.8, starting on p. 655.

Question Explanation:

There is no way to solve this problem other than to simulate the 16 insertions. The following table gives a description of what happens at each step. We show in boldface the first time each key becomes a member of a bucket by itself.

Inserted Key	*i*	*n*	*r*	Changes to Buckets
1111	1	2	1	Bucket 1 = {**1111**}
1110	1	2	2	Bucket 0 = {**1110**}
1101	1	2	3	Bucket 1 = {1101, 1111}
1100	2	3	4	Bucket 0 split into 00 = {**1100**} and 10 = {1110}. Bucket 1 is renamed 01, but continues to hold {1101, 1111}.
1011	2	4	5	Bucket 01 is split into 01 = {**1101**} and 11 = {1011, 1111}
1010	2	4	6	Bucket 10 = {1010, 1110}
1001	3	5	7	Bucket 00 is split into 000 = {} and 100 = {1100}. Buckets 01, 10, and 11 are renamed 001, 010, and 011, but remain unchanged, except that 001 = {1001, 1101}.
1000	3	6	8	Bucket 001 is split into 001 = {**1001**} and 101 = {1101}. Bucket 000 becomes {**1000**}.
0111	3	6	9	Bucket 011 becomes {0111, 1011, 1111}. Note the overflow is OK, and is handled with an overflow block.
0110	3	7	10	Bucket 010 is split into 010 = {**1010**} and 110 = {0110, 1110}.
0101	3	8	11	Bucket 011 is split into 011 = {**1011**} and 111 = {0111, 1111}. Bucket 101 becomes {0101, 1101}.
0100	3	8	12	Bucket 100 becomes {0100, 1100}.
0011	4	9	13	Bucket 000 is split into 0000 = {} and 1000 = {1000}. Buckets 001 through 111 are renamed 0001 through 0111, but retain their previous contents, except that 0011 becomes {0011, 1011}.
0010	4	10	14	Bucket 0001 is split into 0001 = {} and 1001 = {1001}. Bucket 0010 becomes {0010, 1010}.
0001	4	10	15	Bucket 0001 becomes {**0001**}.
0000	4	11	16	Bucket 0010 is split into 0010 = {**0010**} and 1010 = {1010}. Bucket 0000 becomes {**0000**}.

The correct choice is: **c)**

4. In a certain state, license plates have three letters followed by three digits. Records for all automobiles, with the license plate as the key, are stored in a partitioned hash table. The letters each receive two bits of the hash address, and the numbers each receive one bit, so there are $2^9 = 512$ buckets.

Several license plates have been partially recognized. That is, an observer noticed some but not all of the letters and/or digits. The following reports of license plates were made, where * represents a position that was not seen:

1. AB*12*
2. C**345
3. DEF6**
4. GH*7**

For each, how many buckets must we examine to find all the records of automobiles with license plates that might match the report. Identify the correct number from the list below.

a) AB*12* requires us to search 8 buckets

b) AB*12* requires us to search 2 buckets

c) C**345 requires us to search 2 buckets

d) AB*12* requires us to search 64 buckets

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

When a letter is missing, we don't know two bits of the hash function, and when a digit is missing we don't know one bit of the hash function. The number of buckets we must search is 2 raised to the power equal to the number of bits we do not know. Thus, the number of buckets searched for each plate is:

$2^3 = 82^4 = 162^2 = 42^4 = 16$

AB*12*

C**345

DEF6**

GH*7**