لكي لا نكون عرضة لخسارة درجة الواجب جميعا

رجاءا نسخ الحل عن طريق

1. فتح ملف وورد جديد
2. ثم تضليل الحل و نسخه الى الملف الجديد و تغيير ما يلزم
3. و استخدام حفظ باسم

لكي يتم ازالة معلومات الملف القديم

17.28. A file has *r* = 20,000 STUDENT records of *fixed length.* Each record has the following fields: Name (30 bytes), Ssn (9 bytes), Address (40 bytes), PHONE (10 bytes), Birth_date (8 bytes), Sex (1 byte), Major_dept_code (4 bytes), Minor_dept_code (4 bytes), Class_code (4 bytes, integer), and Degree_program (3 bytes).An additional byte is used as a deletion marker. The file is stored on the disk whose parameters are given in Exercise 17.27.
a. Calculate the record size *R* in bytes.
b. Calculate the blocking factor *bfr* and the number of file blocks *b*, assuming an unspanned organization.
c. Calculate the average time it takes to find a record by doing a linear search on the file if
(i) the file blocks are stored contiguously, and double buffering is used;
(ii) the file blocks are not stored contiguously.
d. Assume that the file is ordered by Ssn; by doing a binary search, calculate the time it takes to search for a record given its Ssn value.

(a)
R = (30 + 9 + 40 + 10 + 8 + 1 + 4 + 4 + 4 + 3) + 1 = 114 bytes

(b)
Solution: Blocking factor bfr = floor(B/R) = floor(512/114) = 4 records per block
Number of blocks needed for file = ceiling(r/bfr) = ceiling (20000 / 4) = 5000 blocks

(c) For linear search we search on average half the file blocks= 5000/2= 2500 blocks.
i. If the blocks are stored consecutively, and double buffering is used, the time to read 2500 consecutive blocks

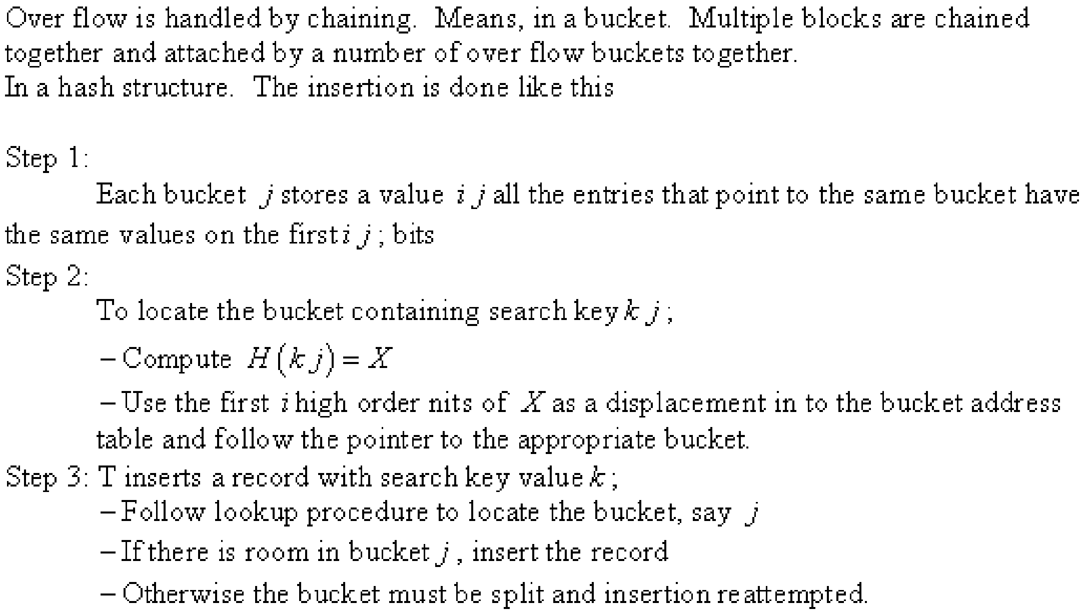
From previous exercise
Rotational delay (rd) = 12.5 msec
Block transfer rate (btr) = 409.6 bytes/msec
Block transfer time (btt) = 1 msec

(k is the number of cylinders required for 2500 blocks)
= ks+rd+(2500*(B/btr))= 5*30+12.5+(2500*(512/409.6))
= 3287.5 msec = 3.2875 sec
(a less accurate estimate is = s+rd+(2500*btt)= 30+12.5+2500*1= 2542.5 msec)

ii. If the blocks are scattered over the disk, a seek is needed for each block, so the time is: 2500 * (s + rd + btt) = 2500 * (30 + 12.5 + 1) = 108750 msec = 108.75 sec

(d) For binary search, the time to search for a record is estimated as:
ceiling(log 2 b) * (s +rd + btt) = ceiling(log 2 5000) * (30 + 12.5 + 1) = 13 * 43.5 = 565.5 msec = 0.5655 sec

17.38. Suppose that we have a hash file of fixed-length records, and suppose that overflow is handled by chaining. Outline algorithms for insertion, deletion, and modification of a file record. State any assumptions you make.



Over flow is handled by chaining. Means, in a bucket. Multiple blocks are chained together and attached by a number of over flow buckets together.
In a hash structure. The insertion is done like this

Step 1:
Each bucket j stores a value $i\ j$ all the entries that point to the same bucket have the same values on the first $i\ j$; bits

Step 2:
To locate the bucket containing search key $k\ j$;
– Compute $H(k\ j) = X$
– Use the first i high order nits of X as a displacement in to the bucket address table and follow the pointer to the appropriate bucket.

Step 3: T inserts a record with search key value k ;
– Follow lookup procedure to locate the bucket, say j
– If there is room in bucket j , insert the record
– Otherwise the bucket must be split and insertion reattempted.

17.43. Suppose we have a sequential (ordered) file of 100,000 records where each record is 240 bytes. Assume that *B* = 2400 bytes, *s* = 16 ms, *rd* = 8.3 ms, and *btt* = 0.8 ms. Suppose we want to make *X* independent random record reads from the file.We could make *X* random block reads or we could perform one exhaustive read of the entire file looking for those *X* records. The question is to decide when it would be more efficient to perform one exhaustive read of the entire file than to perform *X* individual random reads. That is, what is the value for *X* when an exhaustive read of the file is more efficient than random *X* reads? Develop this as a function of *X*.

Solution: Total blocks in file = 100000 records * 240 bytes/record divided by 2400 bytes/block = 10000 blocks.
Time for exhaustive read = s + rd + b*btt = 16 + 8.3 + (10000) * 0.8 = 8024.3 msec
Let X be the number of records searched randomly that takes more time than exhaustive read time. Hence, X(s + r + btt) > 8024:3
X(16 + 8:3 + 0:8) > 8024:3
X > 8024:3=25:1 Thus, X > 319:69
i.e. If at least 320 random reads are to be made, it is better to search the file exhaustively.