Modified

CBCS SCHEME

USN | | | | | | | | | | |

18IS61

Sixth Semester B.E. Degree Examination, June/July 2023
File Structures

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. Differentiate between file structures and data structures. Illustrate the evaluation of file structures. **(08 Marks)**

b. Calculate the space required on the tape if we want to store 1 million 100 bytes records on 7250 bpi tape that has an internal block gap of 0.2 inches and blocking factor of 60. **(06 Marks)**

c. Suppose you are writing a list of names to a text file, one name per write statement. Why is it not a good idea to close the file after every write and reopen it before the next write? **(06 Marks)**

OR

2 a. Illustrate the structure of CD-ROM sector. **(05 Marks)**

b. Write a program to read a series of names one per line standard input and write out those names spelled in reversed order to standard output. Use read IO directions and pipes to do the following:

i) Input a series of names that are typed in from the keyboard and write them out reversed to a file called file 1.

ii) Read the names in file 1, then write them out re-reversed to a file called file 2. **(08 Marks)**

c. Differentiate between constant linear velocity and constant angular velocity. Justify how constant linear velocity is more suitable for audio CD. **(07 Marks)**

Module-2

3 a. How space can be reclaimed from the deletion of records in the variable length records. Illustrate with an example. **(06 Marks)**

b. Discuss the limitation of secondary key index. Explain the linking the list of references technique to overcome the limitation. **(06 Marks)**

c. What is redundancy reduction? Why run length encoding is an example of redundancy reduction? How would we encode the following sequence of hexadecimal byte values? 22, 23, 24, 24, 24, 24, 24, 24, 24, 25, 26, 26, 26, 26, 26, 26, 25, 24. **(08 Marks)**

OR

4 a. Explain the Huffman code algorithm. Generate the Huffman code for CDFFE. **(08 Marks)**

b. What is the difference between internal and external fragmentation? How can compaction effect the amount of internal fragmentation in a file? What about external fragmentation? **(06 Marks)**

c. Write a C++ program on simple index for a file of student objects. Implement add( ) and search( ) function using the index. **(06 Marks)**

Module-3

5 a. Define heap. List the properties of the heap. Build the heap binary tree for the following keys. Show each step clearly. F D C G H I B E A. **(08 Marks)**

b. Explain the merging as a way of sorting of large files on disk. **(08 Marks)**

c. Explain the consequential operations with an example. **(04 Marks)**

OR

6 a. Define B-tree. List the properties of B-tree. Build the B-tree for the given keys. C S D T A M P I B W N G U R K E H O L J Y Q Z F X V. Show each step clearly. **(14 Marks)**

b. Derive the equation for worst case of search depth of B-tree. B-tree of order 512 that contains 1,00,000 keys. Find the maximum depth of the tree. **(06 Marks)**

Module-4

7 a. Illustrate the maintenance of sequence set with an example. **(06 Marks)**

b. Describe the simple prefix B+ tree and its maintenance. **(08 Marks)**

c. List the strength and weakness of B+ trees and B trees. **(06 Marks)**

OR

8 a. Describe file structures that permit each following type of access:
i) Sequential Access only
ii) Direct Access only
iii) Indexed Sequential Access. **(09 Marks)**

b. Explain with a suitable example variable length separator and corresponding index. **(05 Marks)**

c. If the keys EMBRY and FOLKS in the simple prefix in the following Fig.Q.8(c) are deleted from the sequence set node. How the index set in the given diagram is affected by the sequence set deletion.

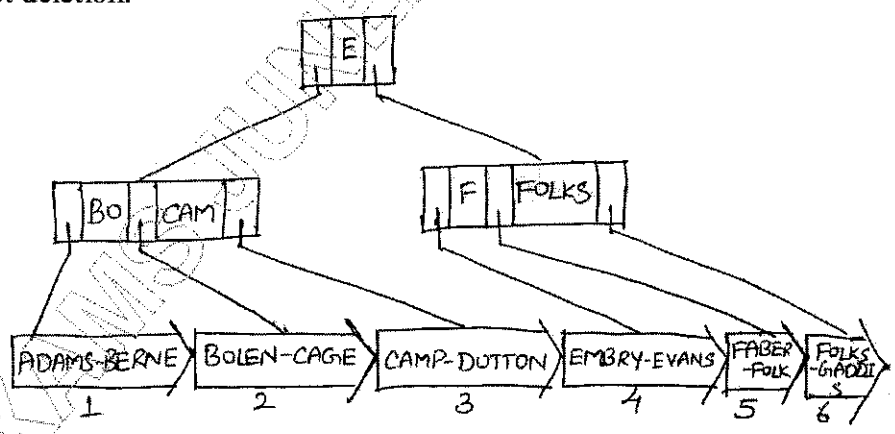


Fig.Q.8(c)

(06 Marks)

Module-5

9 a. Illustrate with suitable diagram dynamic hashing and linear hashing. **(08 Marks)**

b. Explain in detail the working of extendible hashing. **(08 Marks)**

c. Describe space utilization of buckets with suitable example. **(04 Marks)**

OR

10 a. What is hashing? Explain simple hashing algorithm with an example. **(10 Marks)**

b. Suppose that 10,000 address are allocated to hold 8000 records in a randomly hashed files and each address can hold one record. Compute the following values:
i) The packing density for the file.
ii) The expected number of address with no re-assigned to them by the hash function.
iii) The expected number of address with one record assigned (no synonym).
iv) The expected number or address with one or more record and one or more synonym.
v) The expected number of overflow records. **(10 Marks)**

* * * * *

Re: Sir, regarding Modification of scheme and solution

"Nagappa Bhajantri" <bhajan3nu@gmail.com>

July 30, 2023 11:58 AM

To: boe@vtu.ac.in

Respected sir,
Here attached scheme for 18IS61 no need of modifications and
It is updated

With regards

On Thu, 27 Jul, 2023, 12:13 pm , <boe@vtu.ac.in> wrote:

Visvesvaraya Technological University
Belagavi, Karnataka – 590 018.

Scheme & Solutions

Subject Title : FILE STRUCTURES **Subject Code :** 18IS61

Signature of Scrutinizer 18/08/21

Question Number	Solution	Marks Allocated
1.a)	Data Structure: Representation of data item in Primary memory to do storage retrieval operations efficiently File structures: Representation of items at record in any memory. Evaluation: i) Fast data retrieval ii) High amount of work for process data input and maintainance of transactions. iii) Efficient use of storage space iv) Protection for failure or data loss v) Minimizing need for reorganization vi) Security for unauthorized access	8 Marks 3 marks 5 marks
1.b)	$S = n(b+g)$ Where S-Space, b=Physical length of data block, g=length of inter block gaps, n- no. of data block. Given $n = 10,00,000$ $g = 0.2$ $b = \frac{\text{block size}}{\text{tape den}} = \frac{100}{7250} = 0.0137$ $S = 10,00,000\ (0.0137+0.2)$ $= 2,13,700$ inches $= 2,13,700/12 = 17808.3$ feet. If blocking factor increases from 1 to 60 $n = \frac{10,00,000}{60} = 16666.6$ $S = 16,666.6\ (0.0137+0.2)$ $S = 3561.6$ inches $S = 296.8$ feet.	6 marks 3 3

"APPROVED"
Registrar (Evaluation)
Visvesvaraya Technological University
BELAGAVI - 590018

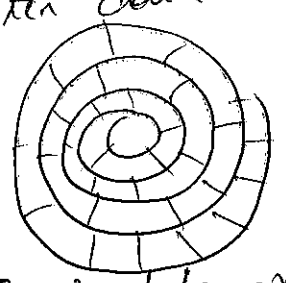
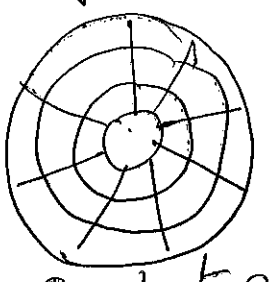
Subject Title : FILE STRUCTURES **Subject Code :** 18IS61

Question Number	Solution	Marks Allocated
1.c)	Opening and closing a file for every single write operation is a bad idea, because i) It is terribly inefficient ii) It requires an extra seek to EOF inorder to append iii) It forfeits atomicity meaning that file may be renamed moved, deleted, returned to or locked by someone else. iv) Repeatedly open and closing of file for every write operation, the chance of failure increases	6 Marks
2a)	Structure of a CD-ROM Sector 12 bytes Synch | 4 bytes Sector ID | 2048 bytes user data | 4 bytes error detection | 8 byte Null | 276 bytes Correction * CD audio uses 16 bits to store each amplitude measurement. * 2 Kilobytes of user data storage * The audio error correction would result in an average of one incorrect byte for every two discs. * The additional error correction information stored within the 2352 byte sector decreases this error rate to 1 uncorrectable byte in every twenty thousand discs.	5 Marks

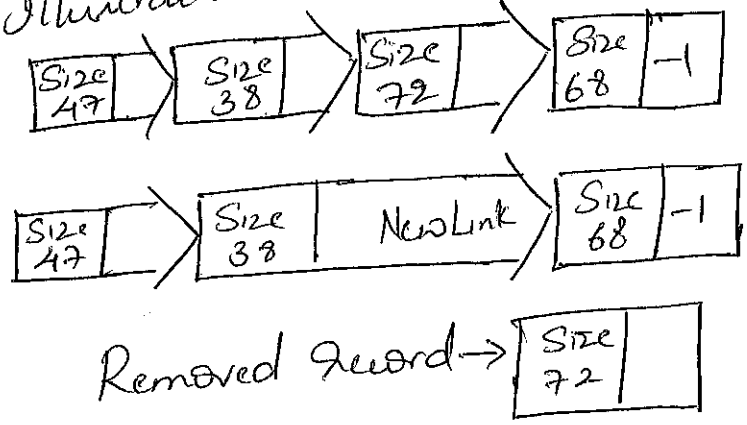
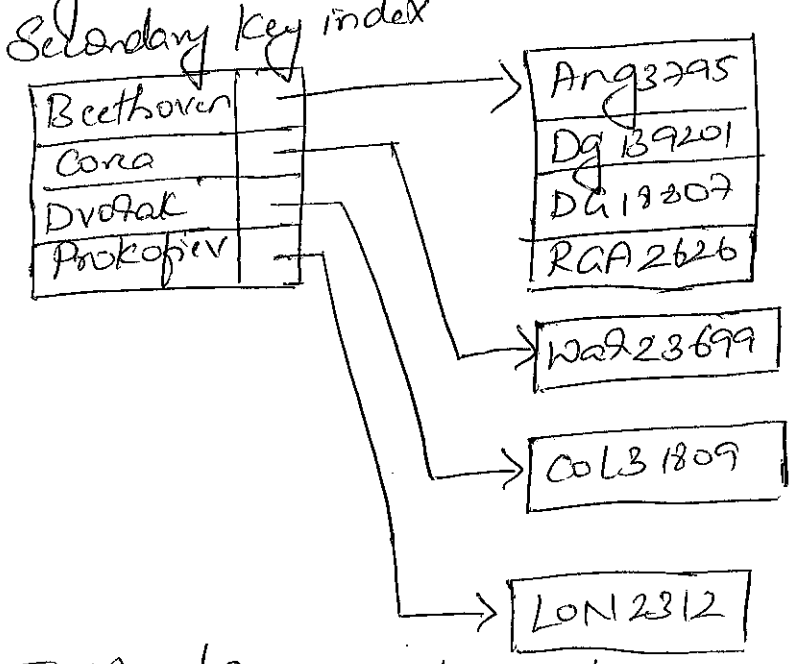
Subject Title : FILE STRUCTURES **Subject Code :** 18IS61

Question Number	Solution	Marks Allocated

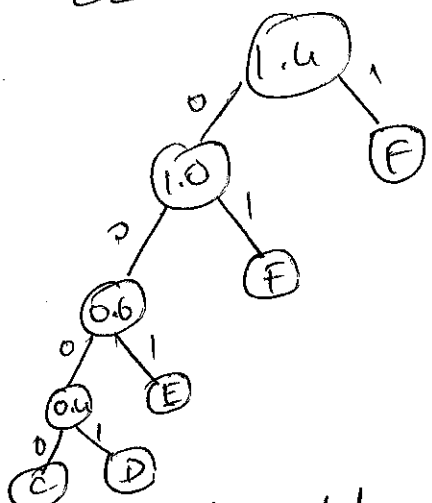
2b)	class Std_file {
	private: char name[10][20], input[20], output[20], str[20];
	public : void std_io();
	void file_io(); };
	void std_file :: std_io()
	{
	int n,j;
	Read the number of names to read
	for (i=0; i<n; i++) gets(name[i]);
	cout << "Reversed name" << endl.
	void std_file :: file_io()
	{ fstream ifile, ofile;
	Read the file containing input names.
	ifile.open(input, ios::in)
	if(!ifile)
	file doesn't exists
	while (!ifile.eof()) {
	ifile.getline(str,20);
	ofile << strrev.
		8 Marks

2.c)	Constant Linear Velocity (CLV) - A sector of the disc takes the same amount of space as a sector towards the center of the disc. Constant Angular Velocity (CAV) with its concentric tracks and pie shaped sectors writes data less densely  Constant Linear Velocity  Constant angular Velocity	7 Marks.

Subject Title : FILE STRUCTURES **Subject Code :** 18IS61

Question Number	Solution	Marks Allocated
3.a)	The method of deleting the record in the list and getting a place a link field. The contents of the fixed length and variable length records and handling are same. Illustration with an example, 	6 Marks.
3.b)	The inverted list - in which secondary key leads to a set of one or more primary keys  Explination of above diagram	6 Marks.

Subject Title : FILE STRUCTURES **Subject Code :** 18IS61

Question Number	Solution	Marks Allocated
3c)	Redundancy reduction is a technique of reducing the duplication. By using a pure binary encoding. By using run length encoding, unused byte value to indicate that a run length code. The hexadecimal byte values, 22 23 ff 24 07 25 ff 26 06 25 24	8 Marks.
4a)	The encoding scheme, Huffman code, determines the probabilities of each value occuring in the data set and builds a binary tree. Huffman code. C D F F E 0.2 . 0.2 0.4 0.4 0.2 C ≤ D ≤ E ≤ F ≤ F  C = 0000 D = 0001 E = 001 F = 01 F = 1	8 Marks. 4 marks. 4 marks.
4b)	The internal fragmentation - The padding in wasted space and cost of using fixed length records, External fragmentation - the space on the avail list rather than being locked inside some other record but is fragmented to be reused.	6 Marks.

Subject title: File Structures Subject Code: 18IS61

The storage compaction can be done by approaches
1) If two records slots on the avail list are physically adjacent, combine them to make a single.
2) To minimize fragmentation before it happens by adopting a placement strategy that the program can use on it selects a record slot from the avail list.

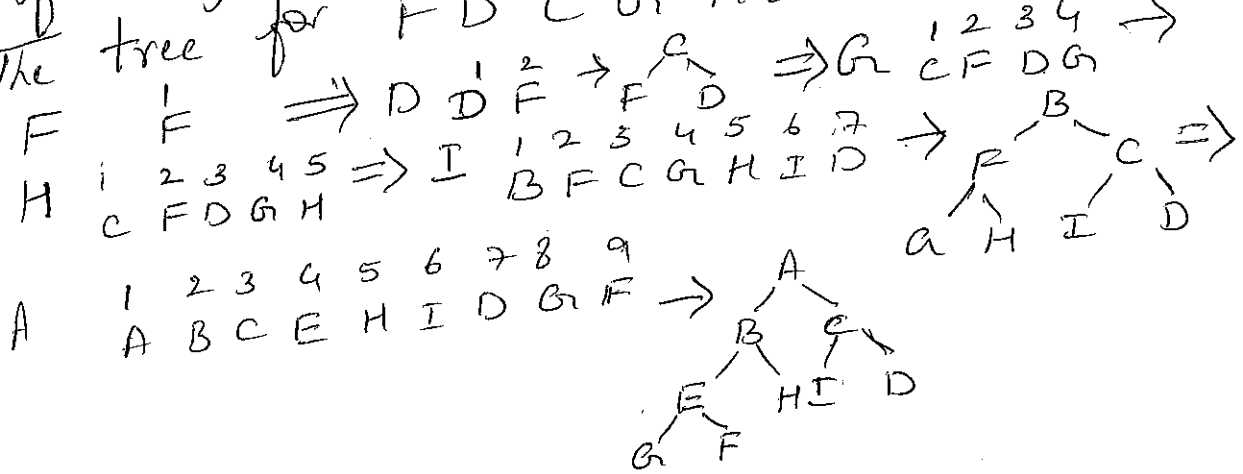
6 Marks

4c)
```
Class student{
  public char name[20],usn[10], age[5], sem[5],
  branch[5]; };
Student s[100], t;
Char buffer[50], buffer1[50];
Char tempusn[15], temprrn[10], keyusn[15];
int count=0,i; fstream fp1,fp2;
void search(){
 Student x;
 fp1.open("std.txt", ios::in);
 fp2.open("ind.txt", ios::in);
 cout<<"Enter USN"; cin>>keyusn;
 for(i=0; i<count; i++)
 { fp2.getline(buffer, 100);
   sscanf(buffer, "%[^|]|%[^|]|", tempusn,
     temprrn);
   If (strcmp(tempusn, keyusn)==0)
     Print the record
 }
void add(){
 cout<<"Enter the name\n"; cin>>t.name;
 cout<<"Enter the usn\n"; cin>>t.usn;
 cout<<"Enter the age\n"; cin>>t.age;
 cout<<"Enter the sem\n"; cin>>t.sem;
 cout<<"Enter the branch\n"; cin>>t.branch;
  pack(t); }
```

Subject title: File Structures Subject code: 18IS61

5a) A heap is a binary tree. The properties are 8 Marks
* Each node has a single key, key greater than or equal to the key at its parent node.
* Complete binary tree - all of its leaves are no more than two levels and keys on lower level are in the leftmost position.
* Storage for the tree can be allocated sequentially as an array in such a way that the root node is i and left & right children of node i are 2i and 2i+1 respectively. Conversely, the index of the parent of node j is $\lfloor j/2 \rfloor$.

The tree for F D C G H I B E A

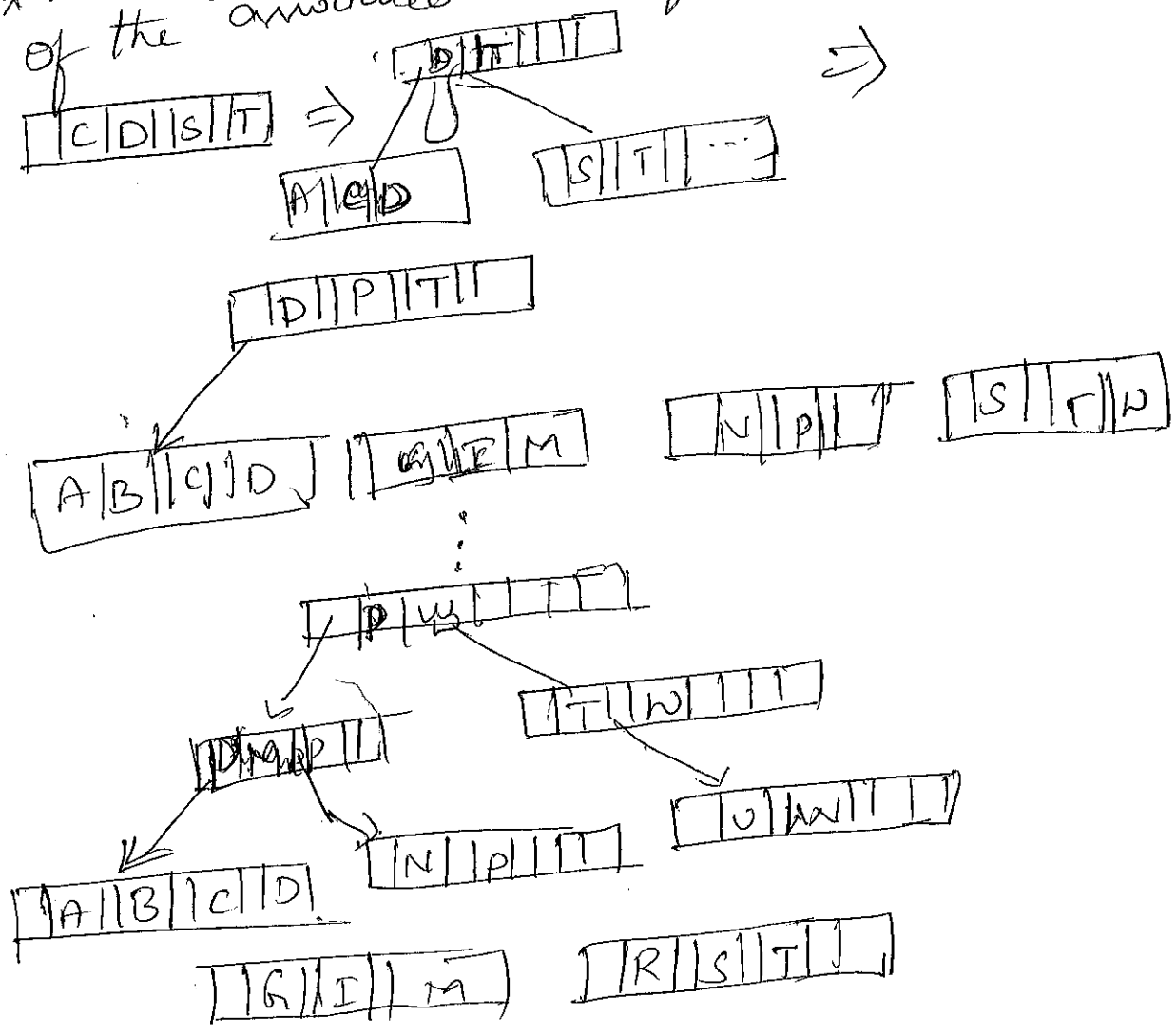


5b) The limitations of keysorting lead to the multiway merge algorithm. 8 Marks
* a sorted subset is created and records back to disk as a sorted subfile.
* Sort large files and can be extended to files of any size.
* Reading of the input file during the run-creation step is sequential and hence is much faster than input.
* During the sort phase: Reading all records into memory for sorting and forming runs and writing sorted runs to disk.
* During the merge phase: Reading sorted runs into memory for merging and writing sorted file to disk.

Subject title: File Structures Subject Code: 18IS61

5.c) The class Consequential Process supports processing of any type of list in the runtime way. DoBuffer supports Buffer operations on any type of buffer. 4 Marks
- It includes operations to match and merge lists.
- It defines list processing operations required for consequential processing as virtual methods.

6.a) Definition of B-tree. 10 marks
the properties are,
* Every page has a maximum of m descendants
* except for the root and the leaves, has at least $\lceil m/2 \rceil$ descendants.
* root has at least two descendants
* All the leaves appear on the same level
* The leaf level forms a complete, ordered index of the associated data file



Subject title: File Structures Subject code: 18IS61

6.b) For any level d of a B-tree, the minimum number of descendants extending from that level is $2 \times \lceil m/2 \rceil^{d-1}$ 6 Marks

For a tree with N keys in its leaves, we can express the relationship between keys and the minimum height d as

$$N \geq 2 \times \lceil m/2 \rceil^{d-1}$$

Solving for d,

$$d \leq 1 + \log_{\lceil m/2 \rceil} (N/2)$$

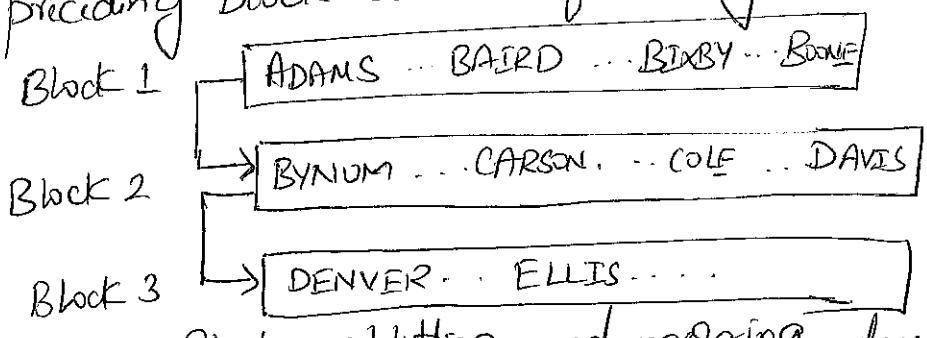
a tree of order 512 that contains 1000000 keys,

$$d \leq 1 + \log_{256} 500000$$

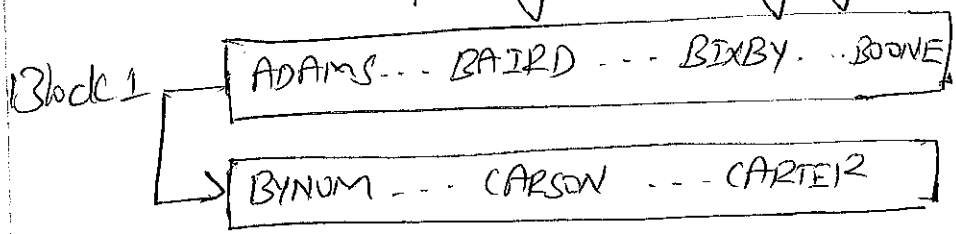
$$d \leq 3.37$$

7a) The ordered set of records as a sequence set. 6 Marks.

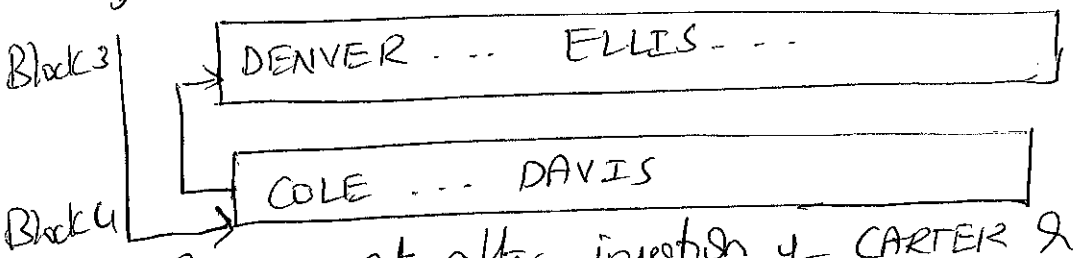
* The best way to restrict the effects of an insertion or deletion to just a part of the sequence set involves a tool.
* The link fields in each block that point to the preceding block and the following block.



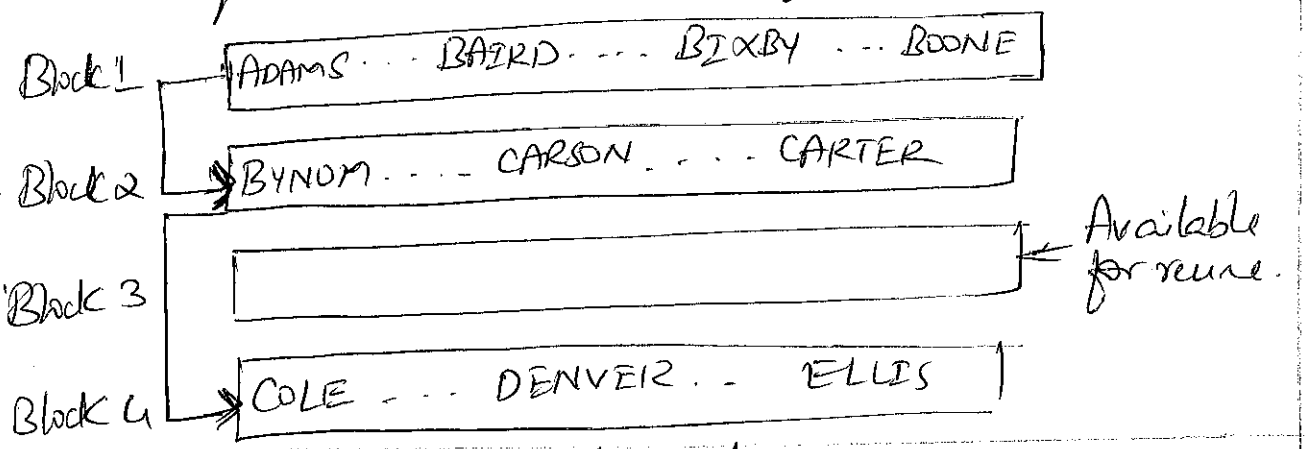
Block splitting and merging due to insertions.



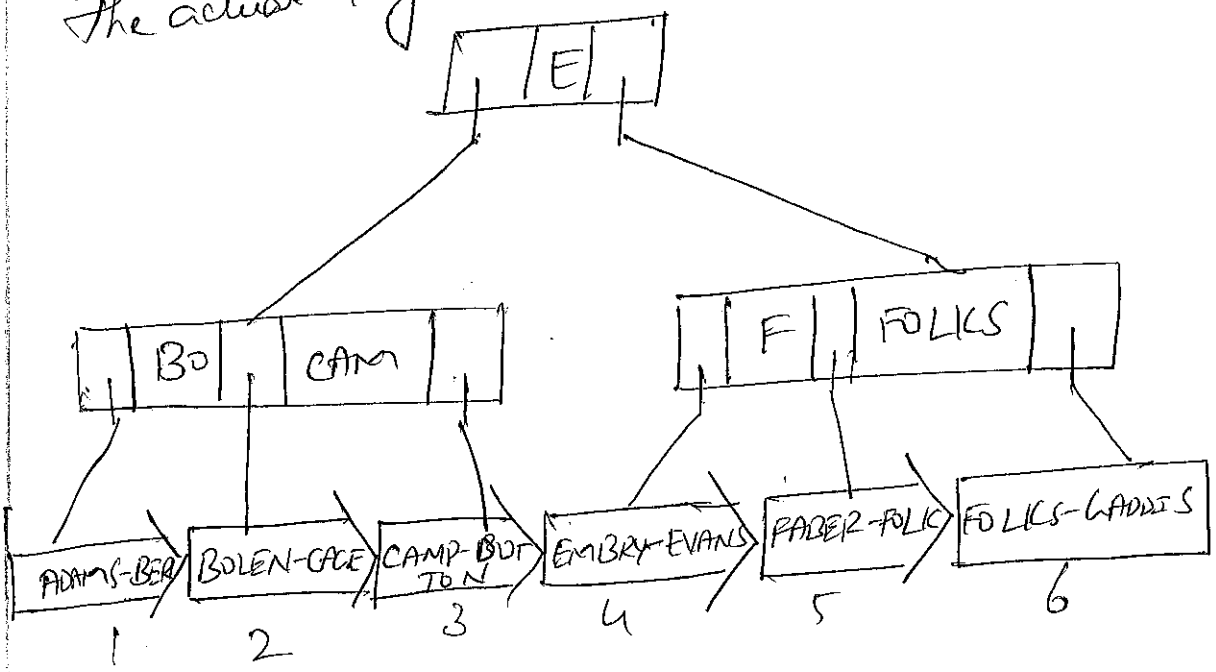
Subject title: File Structures Subject Code: 18IS61



Sequence set after insertion of CARTER record



7.b) The B-tree index is called the index set. With the sequence set, it form a file structure. The modified simple prefix indicates that the index set contains shortest separators or prefixes of the keys rather than copies of the actual keys. 8 Marks



Subject title: File Structures Subject Code: 12IS61

6 Mark

7c)

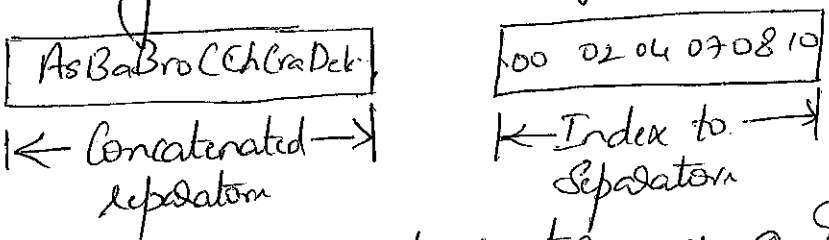
Strength	Weakness
B+ trees	
1. Record can be fetched in equal number of disk access	1. Deletion operation may result in case 1: if both leaf nodes & index nodes doesn't go below the fill factor Case 2: If the leaf node goes below fill factor and index node doesn't go below the fill factor.
2. Height of the tree remains balanced and less compared to B trees.	
3. Keys are used for indexing	
4. Can access the data stored in B+ trees Sequential / directly	
5. Faster Search Query as data stored on leaf nodes.	
B-trees	
1. Keeps key in sorted order	1. Leaf and non leaf nodes are of different size (Complication for space)
2. Uses hierarchical index to minimize number of disks read	2. Deletion may occur in a non leaf node.

8a) 9 Mark

Sequential access must start at the beginning and access each element inorder.
For ex: magnetic tape - only Sequential access.
Direct access allows the access of any element directly by locating it by its index number or address.
For ex: CD had direct access.
Record can be accessed randomly if the primary key is known. Index file is used to get the address of a record and then the record is fetched from the data file.

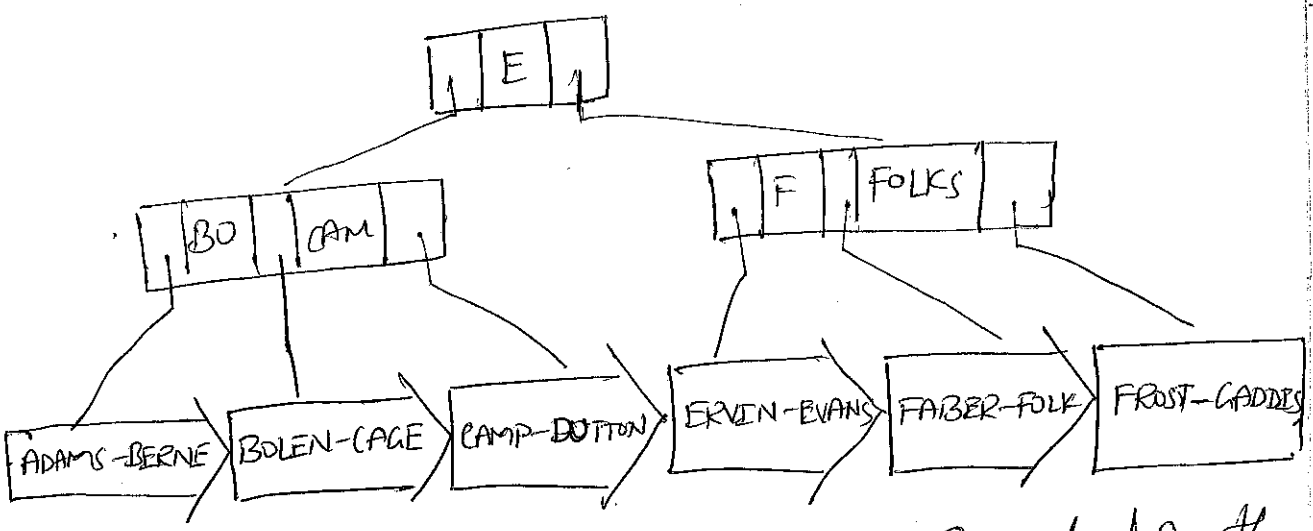
Subject title: File Structures Subject Code: 18IS61

8.b) Variable length separator - The index consists of fixed length references, binary searching on the index. Retrieving the variable length records. 5 Marks



* References are made in terms of a relative block number (RBN) - analogous to a relative record number.
Separator count: Find the middle element in the index to the separators to binary search begin.

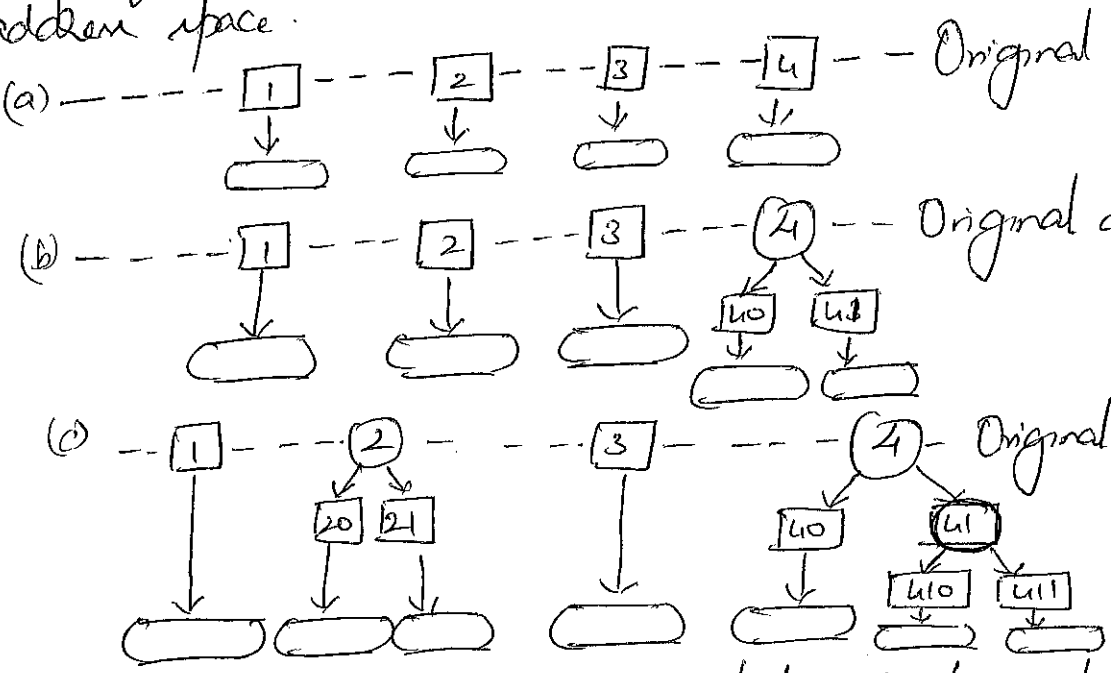
8.c) 6 Marks



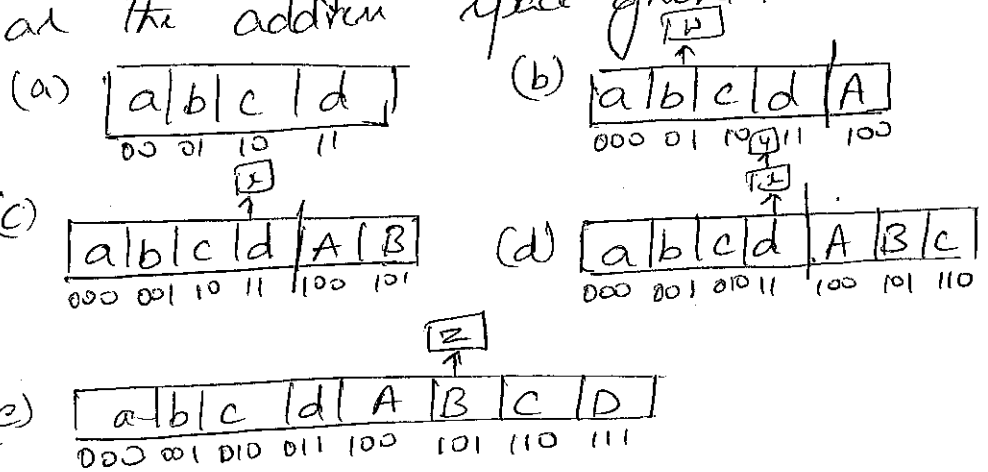
* The deletion of the EMBRY and FOLKS records from the sequence set leaves the index set unchanged
* There is no merging or redistribution, the effect is limited.
* The number of sequence set blocks is unchanged and since no records are moved between blocks
* E is still a perfectly good separator for sequence set blocks 3 and 4, so there is no reason to change it in the index set.

Subject title: File Structure Subject Code: 18IS61

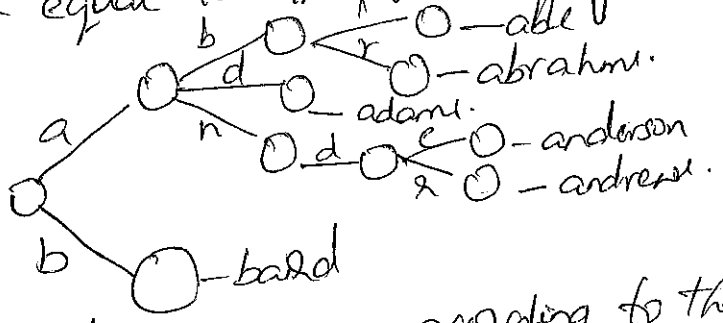
9.a) Dynamic hashing scheme involve the use of two hash functions rather than one – covers original address space. 8 Marks



Linear hashing – uses more bits of hashed value as the address space grows.



9b) Extendible hashing is to combine conventional hashing with another retrieval approach called the trie. Radix search – branching factor of the search tree is equal to the number of alternative symbols. 8 Marks



indexes names according to the letters of the alphabet.

Subject title: File Structures Subject Code: 13IS61

9.c) Space utilization of buckets. 4 Mark

$N = \frac{r}{b \ln 2} N$ — ratio of actual number of records to the total number of records.

Utilization $= \frac{r}{bN} = \ln 2 = 0.69$.

As the buckets fill up the space utilization can reach part 90 percent.

10 Mark

10.a) Hashing - definition. $a = h(k)$ Where a = home address, h = hash function, k = key 2 Mark

Hashing algorithm. 3 Mark

Step 1: Represent the key in numerical form

Step 2: Fold and Add

Step 3: Divide by a prime number and use the remainder as the address. P

Examples to be given for each algorithm 5M

10.b) i) Packing density for the file $= \frac{n}{r} = \frac{8000}{10000} = \frac{4}{5} = 0.8$ 10 Mark

Packing density = 0.8

ii) Expected number of address $= NP(0) = \frac{10,000 \times (0.8)^0 \times e^{-0.8}}{0!}$

$= 4504.5$

Subject title: File Structures Subject Code: 18IS61

iii) Expected number of record with one record assigned

$$NP(1) = \frac{10,000 \times (0.8)^1 \times e^{-0.8}}{1!}$$

$$= 3603.6$$

iv) Expected number of address with one or more record and one or more synonym

$P(2) = 0.144$
$P(3) = 0.038$
$P(4) = 0.0076$
$P(5) = 0.0012$

$= P(2) + P(3) + P(4) + P(5)$
$= 0.144 + 0.038 + 0.0076 + 0.0012$
$= 0.1908 \times 10,000$
$= 1908$

v) Expected number of overflow records

$= 1 \times 10,000 * 0.144 + 2 \times 10,000 \times 0.038 +$
$3 \times 10,000 \times 0.0076 + 4 \times 10,000 \times 0.0012$
$= 1440 + 760 + 228 + 48$
$= 2476$

"APPROVED"
Registrar (Evaluation)
Visvesvaraya Technological University
BELAGAVI - 590018

x — x — x