University of Toronto
csc228S - File Structures and Data Management, Spring 2001

Midterm Solutions — Afternoon Section

Friday 02 March 2001

Duration: 50 minutes

Aids allowed: None

Family Name: ____________________ Given names: ____________________

Student #: ____________________ Tutor: ____________________

- There are 6 pages, including this one. The test is out of 30 marks and the value of each question is provided; please use this information to manage your time effectively.
- For questions that involve writing code, comments are not necessary. If you need to call a standard function but can't remember the correct order of arguments, just indicate the meaning of each argument.

Part A:	______ / 8
Part B:	______ / 4
Part C:	______ / 6
Part D:	______ / 2
Part E:	______ / 3
Part F:	______ / 3
Part G:	______ / 4
Total	______ / 30

Part A [8 marks in total]

Suppose we want a file that contains a linked list of integers. The file format is as follows:

- The file begins with a long integer, written in binary, which is the byte offset of the first node in the linked list.
- This is followed by a series of nodes, in no particular order. Each node contains:
  - an int, written in binary, which is the data value for that node.
  - a long integer, written in binary, which is the byte offset of the next node in the linked list (or -1 if this is the last node).

Below is the declaration for a suitable node class.

```
class Node {
public:
    int data;
    long next;

    // Write me to 'fs', at the current position of the seekp pointer.
    // Return where it is that I've been written.
    long writeMe(ostream &fs);

    // Read into me the Node that is 'where' bytes from the beginning
    // of 'fs'.
    void readMe(istream &fs, long where);

};
```

Complete these functions:

```
long Node::writeMe(ostream &fs){

    long where = fs.tellp();
    fs.write(&data, sizeof(int));
    fs.write(&next, sizeof(long));
    return where;

}
```

```
void Node::readMe(istream &fs, long where){

    fs.seekg(where);
    fs.read(&data, sizeof(int));
    fs.read(&next, sizeof(long));

}
```

Part B [4 marks in total]

For this question, assume we are talking about the kind of B-tree used in lecture and in our textbook. Also assume that the height of a tree is the number of nodes on the longest path from the root to a leaf (including the root and the leaf).

Suppose we have a B-tree of order M and height h.

What is the minimum number of keys that this tree might have at the leaf level? $\underline{2\lceil M/2\rceil^{h-1}}$

What is the maximum number of keys that this tree might have at the leaf level? $\underline{M^h}$

Part C [6 marks]

Suppose we have a file containing n variable-length records, plus f records on the free list. Fill in the table below to show the big-oh time complexity of each of the following operations, in the worst case, depending on how we organize the free list.

	insertion into free list	deletion from the free list
first-fit (keep free list unsorted)	$O(1)$	$O(f)$
best-fit (keep free list sorted small to big)	$O(f)$	$O(f)$
worst-fit (keep free list sorted big to small)	$O(f)$	$O(1)$

Part D [2 marks]

Suppose `sizeof(int)` is 6 and `sizeof(char)` is 2.

Fill in the assignment statement below so that writing `n` to a file as text would consume more file space than writing it in binary.

```
int n = and integer with more than 3 decimal digits.
```

Fill in the assignment statement below so that writing `n` to a file in binary would consume more file space than writing it as text.

```
int n = Any integer with fewer than 3 decimal digits.
```

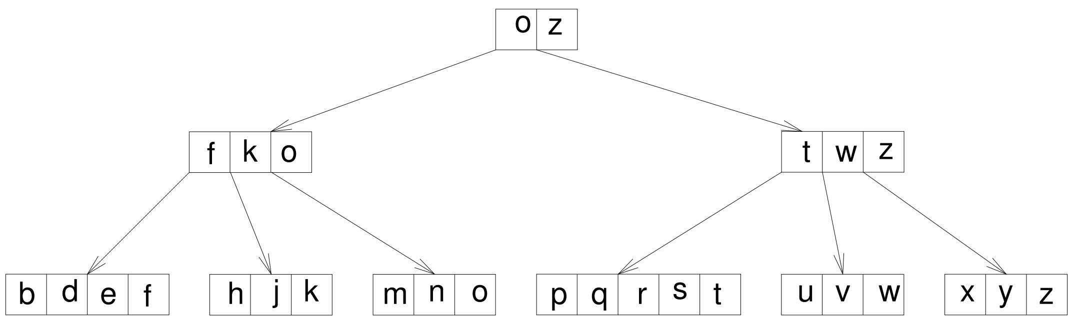
Part E [3 marks; -1 for each wrong answer]

Below are three separate sets of codes. For each set, circle YES or NO to indicate if the set is instantaneous.

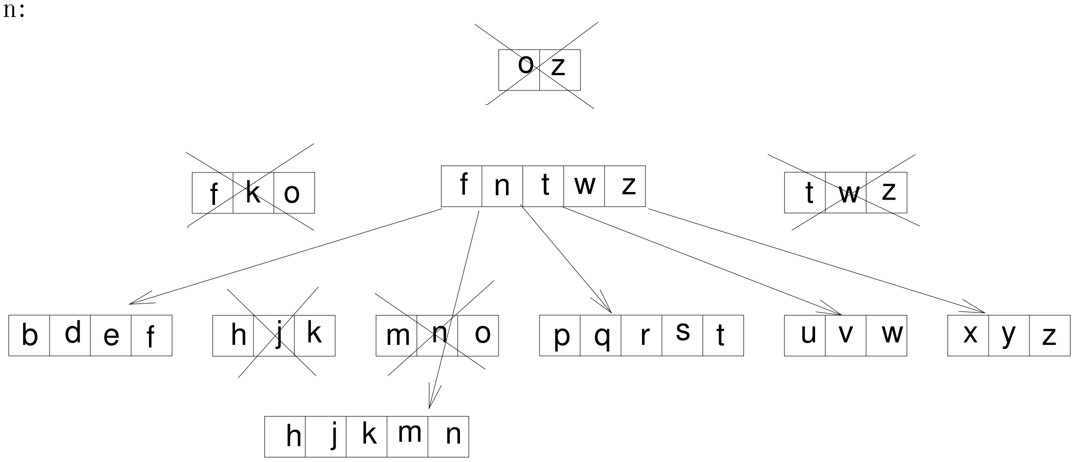
101, 01, 1001, 1000, 110, 111	$\boxed{\text{YES}}$ / NO
11, 10, 01, 00	$\boxed{\text{YES}}$ / NO
111, 110, 01, 00, 100, 1101, 1011	YES / $\boxed{\text{NO}}$

Part F [3 marks]

Below is a B-tree with order 5 (*i.e.*, $M = 5$). Show what would happen if you deleted the key `o`. You may write your answer directly on the tree. Be sure to cross out any values or nodes that are deleted.



Solution:



Part G [4 marks in total]

You are designing a new operating system which will have inodes similar those in Unix but with some variations. The maximum file size in your operating system will be 2^{24} blocks. You are considering an inode with 10 direct pointers, 1 pointer to a 1-level index and 2 pointers to a 2-level index. Your blocks are 2^{10} bytes and you want to use 4-byte pointers.

Suppose the inode is in memory but none of the index blocks are. Counting the first block of the file as block number 1, which is the first block in the file that requires exactly 3 seeks in order to be read?

ANSWER: Block number $\underline{10 + 2^8 + 1}$

`Explanation: Each index block can store` $2^{10}/4 = 2^8$ `pointers. The first 10 blocks of the file are accessible in 1 seek directly from the inode. The next` 2^8 `are accessible in two seeks via the 1-level index. The next block after that must be accessed in 3 seeks via the 2-level index.`

What is the last block of the file that requires exactly 3 seeks in order to be read?

ANSWER: Block number $\underline{10 + 2^8 + 2^{16} + 2^{16}}$

`Explanation: The 2-level indexes give access to file blocks in 3 total seeks. The very last block accessible by the second two level index is the last one accessible in 3 seeks (and is also the last block of the file, since there is no 3-level index). Each two-level index gives access to` $2^8 * 2^8 = 2^{16}$ `file blocks.`

There is a problem with your design. What is it?

`It doesn't provide enough pointers to access` 2^{24} `file blocks as desired.`

Explain your answer, using a relevant equation.

`As we saw, the total number of file blocks that can be pointed to by this inode is` $10 + 2^8 + 2^{16} + 2^{16} = 10 + 2^8 + 2^{17} <<< 2^{24}$.

End of Examination