Some notes on DM2 and the current text.
from the Bellenot

1 "Prove things" it seems that the CS folks most want out of successful DM2 students is the ability to prove things. Hence I started with chapt. 9 (After a 1 week introduction to "practical logic" (I have something written up if you would like to look at it or use). Mott and Nichols started elsewhere.

to make the proofs last longer I do chapter 4 to fill in time starting in 3rd (2nd?) week.

2. Enclosed you will find a list of theorems, lemmas and the like which you may find useful. You will note the very step by step nature, the students aren't very strong

3. Induction proofs are even more important to CS than Math. Hence some strange "theorems" in the list.

4 Chapter 4

A. §4.1 — this is (in my opinion) the hardest in the text. Problems I think I have solved sometimes prove to be wrong. Personally I do 4.2—4.4 first so that you can use systems of equations

B. §§4.2-3 Just like ODE's. But there isn't enuff problems like solve $a_n - 5a_{n-1} = 2$ in text (Bellenot & Mott have additional problems, but there may errors still). Here you find out how weak their Algebra is (Mott found 3 students who hadn't taken Alg-Trig in his class)

I do these via "Cook book" Mott does 4.4 first to motivate this

C. §4.4. Also needs more problems

there are errors too!

D Here the text misses the whole point. Consider

$$a_n = 2a_{n/3} + n^2 \qquad a_1 = 3$$

Let $n = 3^k$, $b_k = a_{3^k}$ then

$$b_k = 2b_{k-1} + 9^k \qquad b_0 = 3$$

This has a solution like $b_k = A \cdot 2^k + B \cdot 9^k$

so $a_n = A \cdot 2^{\log_3 n} + B \cdot 9^{\log_3 n}$

$= A \cdot n^{\log_3 2} + Bn^2$

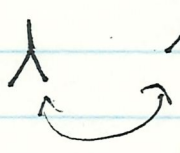
These are more important than you might think. Divide & Conquer is a very useful thing in C.S. see MERGE SORT (§9.5)

5 Chapter 5 we do 5.1 & 5.2 only. Here you will find out how weak the students are in Combinatorics Be Careful not to assign all the easy problems at once.

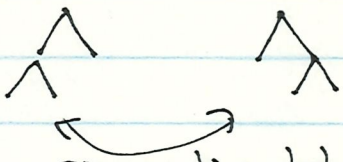
6. Chapter 9 – I consider this the heart of the course. There are several weaknesses of the text

A. There are three kinds of TREE's

1. Tree – conn. graph w/ no circuit
2. ~~Ordered~~ Directed Tree – a tree w/ root & edges directed away from root. The terms FATHER, SON etc have meaning
3. Ordered Tree – a directed tree w/ Left to right ordering. The terms RIGHTSON etc have meaning



same tree different direded trees



same directed but different ordered trees

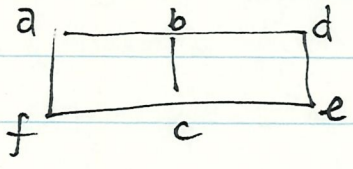
B I carefully stay away from the terms in 9.1 until

D Otherwise §9.1 is O.K

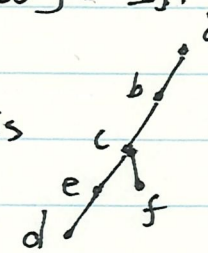
7 §9.2 — the same can't be said for this section. The Problems are also bad

A. The main points are DFS (depth-first search) & BFS (breadth FS) I introduce these via spanning trees (§9.4) for conn. graphs. These are two ways of checking all possible vertices. DFS is perhaps more important.

B. To make it easier to grade I insist that DFS & BFS goes to the "lowest" vertex possible & the same for its root. [left to right too]. If

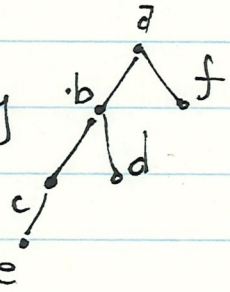


then DFS is



and no other

BFS yields uniquely



C. I have a hard copy of a program written to solve Ex 18

8§9.3 The traveling Salesman Problem uses BRANCH AND BOUND — put it off til the end. If you are lucky you will run out of time

9§9.4 Spanning trees are in 7 above.

Preorder, Postorder & Inorder are important. Most if not all of your students will already have taken PASCAL which supports recursion. Hence, the following

```
PROCEDURE PREORDER(X:NODE)

    BEGIN
      IF X ISN'T A LEAF THEN BEGIN
        WRITE X;
        PREORDER (X.RIGHTSON);
        PREORDER (X.LEFTSON);
      END;
    END

BEGIN (* MAIN PROGRAM *)
    PREORDER (ROOT)
END
```

Inorder goes RTson ; X , LTson
Postorder goes Rtson , Ltson , X

A. An interesting application is changing usual algebraic expressions (infix (ie. in inorder)) to Postorder (Postfix or reverse Polish) This is the way most compliers do this sort of thing. Postfix is easy to evaluate using a "stack"

B. Exercise 25 is similar to an example in §4.4.

10 §9.5

A. O(f(n)) isn't done well in the book. There are things written up. I've supplement on big O here time via L'Hopital's rule. I had only one student in DM2 who hasn't already passed Calculus 1

B. The Theorem on p. 310 is interesting as is Ex 3

11 Chapt 10 Algorithms — examples there of Shortest Path, Prims, Kruskals are all easy to get across. Floyd's requires too much work if you are not a computer.

A §10.3 Is very nice and takes at least a week to get across. Again more problems are needed. Variations (In Example 4 & 5) are O.K but stay away from Example 7 unless you are trying to kill time.

B. §10.4 is also nice althought I short-change the Augmenting Matching Algorithm.

12 Algorithms can be classed into one or more of the following classes.

GREEDY — Prim's, Kruskals
OPTIMAL IS LOCAL OPTIMAL — Shortest Path
DIVIDE & CONQUER — Merge Sort.
~~SEAT~~ BFS — find all solutions
DFS — find a solution
BRANCH & BOUND — traveling Salesman
ALL OTHERS

The main class this is a prereq for is Data Structures & Algorithms

13. There exist lots of copies of texts etc if you want them.