

**B. Math III Year**

**2006 Mid-term Examination – CS III (Max. marks: 40 (weight 20%) Time 3 hours)**

**Q1 [15 Marks: 1.5 Marks for each part]** Give very brief answers

- a. State true or false with a brief reason: “Merge sort runs faster if the input is already sorted”
- b. What is the worst case runtime for the Quick sort algorithm? Why?
- c. What are the best case, worst case, and average case order for the radix sort algorithm?
- d. What is the order of inserts and deletes on arrays (assume that array has n elements)?
- e. When deleting a BST node that has 2 non-NULL children, which rule of thumb could one use to rearrange the remaining tree?
- f. Which data structure would be most appropriate for representing path of a traveling medical representative who will take the reverse path on the return trip?
- g. What is the order of the Quick Sort algorithm if the largest value in the list is always chosen as the pivot?
- h. Data types are important for a number of reasons. We discussed four reasons in the class. State these four reasons (use one word to describe each reason).
- i. What is the main reason for recursive algorithms for functions such as Fibonacci numbers to be slow?
- j. The decision to use recursive vs non-recursive methods usually lead to which trade-off?

**Q2 [5 Marks]** This question refers to the code in the adjoining table:

There are some errors (logical as well as syntax) in this code. Identify and correct them (refer to the line numbers in your answer).  
What is this program attempting to do?

1. #include <stdio.h>	9. do_something(S)
2. main()	10. {
3. {	11. if (S != NULL)
4. char S[80];	12. {
5. printf("Enter text:");	13. do_something(S++);
6. gets(S);	14. putchar(*(--S));
7. do_something(S)	15. }
8. }	16. }

**Q3 [8 Marks]** The complexity for many divide and conquer algorithms is given by a recurrence relation:

$$T(n) = T(1) \text{ for } n=1 \text{ and } T(n) = aT(n/b) + f(n) \text{ for } n > 1$$

where a and b are known constants, T(1) is known, and n is a power of b. Obtain a closed form expression for T(n) as a function of n, f(n) and T(1).

**Q4 [5 Marks]** Construct a binary tree for the following preorder and inorder sequences:

Preorder: a b c e i f j d g h k l

Inorder: e i c f j b g d k h l a

**Q5 [7 Marks]** Write an algorithm to determine if a linked list contains a cycle in it, and, at what node the cycle starts. State the order of your algorithm (Marks are given for the correct data structure declaration, correct logic, order of the algorithm, and a proper pseudo-code convention.)