Indian Statistical Institute
B.Math I Year
End-Semester Examination for Computer Science -I
Backpaper

Time: 3 hours Marks: 100

Note: Answer all the questions. The marks for each question are indicated in square brackets after the question.

1. Insertion Sort:

Recall the insertion sort algorithm: suppose you are given n cards on a table. Move one card at a time from the table to your hand such that after every move the cards in your hand are sorted. In other words, the card that you pick from the table is placed at its "appropriate place" in the hand. After moving the last card, it is clear that the set of cards in your hand is sorted.

(a) Derive a formula for T(n), the worst-case computational complexity of insertion sort. [10]
(b) Write a C program that sorts a set of integers in increasing order, using insertion sort. [15]

2. Linear Search:

Linear search is a very straightforward algorithm to search for an element v in an unsorted set A[1..n]: compare A[k] and v for k = 1,…n, and report the number of values of k for which v = A[k]. If this number is greater than zero, then A[1..n] contains v, otherwise it does not.

(a) What is T(n), the computational complexity of linear search? [5]
(b) Write a C program to implement linear search. [10]

3. Sieve of Eratosthenes:

This is a very old method, due to Eratosthenes (276-196 BC), to list prime numbers, and it goes like this. First list all natural numbers from 2 onwards:

```
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 …
```

Then mark all multiples of 2:

```
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 …
    x   x   x    x     x     x     x     x     x
```

Move to the next unmarked number, which in this case is 3, then mark all its multiples:

```
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 …
    x   x   x x  x     x     x  x  x     x     x
```

Continue in this fashion, marking all multiples of the next unmarked number until there are no new unmarked numbers. The numbers which survive this marking are primes. Design a C program to make a list of primes less than N using Eratosthenes' method. [20]

4. Prime Factorization of an Integer:

One way of factoring an integer n into primes is by trial division. This algorithm can be described by the following pseudo code:

```
Begin: given a positive integer n

  Set d = 2   // the trial divisor
  While n > 1,
    If d divides n,
    then
       write down the factor d
       replace n by n/d
    else
       replace d by d + 1

  go to Begin
```

(a) Describe how this algorithm works for n = 60. [5]
(b) Implement the algorithm in a C program. [15]

5. The 3n+1 Problem:

Consider the following algorithm:

```
1.              input integer n>0
2.              print n
3.              if n = 1 then STOP
4.                 if n is odd then replace n by 3n+1
5.                 else replace n by n/2
6.              GOTO 2
```

For example, if the input is 5, then the numbers printed will be 5 16 8 4 2 1. The number of numbers printed for any input integer is called its *cycle length*. In our example of 5, the algorithm terminates, and the cycle length is 6.

(a) Given the input 22, what is the sequence of numbers printed? [5]
(b) For any two numbers *i* and *j* write a C program to determine the maximum cycle length over all numbers between (and including) *i* and *j*. [15]