Algorithms Homework 2

University of Virginia

Gabriel Robins

Please make all algorithms as efficient as you can, and state their time and space complexities.

1-17. Solve the following problems from the [Cormen, 2nd Edition] algorithms textbook:

p. 37: 2.3-7
p. 50: 3.1-1, 3.1-4, 3.1-7
p. 57: 3.2-4, 3.2-5
p. 58: 3-2, 3-3
p. 85-88: 4-1, 4-2, 4-4, 4-6
p. 94: 5.1-3
p. 98: 5.2-3, 5.2-4
p. 105: 5.3-3
p. 117: 5.4-6

18. What is the minimum number of N-sorter circuits (i.e., an N-input, N-output device where the outputs are the sorted permutation of the inputs) needed to construct a (2N)-sorter?

19. Give a sub-linear algorithm for finding the median among all the elements in two given (equal-sized) sorted arrays?

20. What is the minimum (exact) number of multiplications and additions needed to evaluate an arbitrary N-degree, one-variable polynomial with integer coefficients?

21. Give a linear-time algorithm that for a sequence of N integers, finds the contiguous subsequence with the largest possible sum.

22. Give an O(N log K)-time algorithm to merge K given sorted lists into a single sorted list, where N is the total number of items in all the lists.

23. Consider the following code for randomly permuting an array A of size N:

```
For i = 1 to N Do Swap(A[i], A[Random(1,N)])
```

where Random(i,j) returns a random integer uniformly distributed in the range i through j, inclusive.

a) Prove that in general, not all permutations are equally probable here.
b) Give an alternate random permuter without this flaw.

24. Give a linear-time (and linear-space) algorithm to sort the ratios of N given pairs of integers between 1 and N. I.e., we need to sort within O(N) time N pairs of the form (a_i,b_i) where $1 \le a_i \le N$ and $1 \le b_i \le N$ using the sort key a_i/b_i.

25. Write a program that when executed, prints out **exactly** itself and stops. No run-time input whatsoever is allowed to be used by the program (i.e., the program may not read the keyboard, a file, RAM, pipes, etc.) You may use any programming language you wish. Note that the program must print itself out exactly, right down to the last punctuation mark, tab, and carriage return. (Your solution should not be a blank program.) Name some applications of this idea.