Homework 8

22C:44 Algorithms, Fall semester 2000

Four problems, ten points each. Due on Thursday November 16.

1 Find Huffman-code for encoding symbols that have the following frequencies. What is the expected number of bits per symbol using your Huffman code ?

Symbol	Frequency
a	0.10
b	0.01
c	0.04
d	0.15
e	0.03
f	0.20
g	0.07
h	0.09
i	0.11
j	0.04
k	0.16

2 F.Gump is a good runner. He can run up to 50 miles uninterrupted, but then he needs to stop and eat a box of chocolates. He wants to run across America along a certain route. He knows the locations of the stores along the route selling his favorite chocolates. He wants to stop as few times as possible. Design a linear time $O(n)$ algorithm for choosing where to stop, where n is the number of chocolate shops along his route. The input to your algorithm is a sorted array $A[1 \dots n]$ of distances of the chocolate shops from the starting point. The last chocolate shop location $A[n]$ is the end point of the route.

Argue why your algorithm is correct.

3 Let n and m be positive integers with $m < n$. Then $r = \frac{m}{n}$ is a rational number between 0 and 1. Egyptian number representation of r is a sum

$$r = \frac{1}{n_1} + \frac{1}{n_2} + \dots \frac{1}{n_k}$$

where all n_i are pairwise different positive integers, that is, $n_i \neq n_j$ when $i \neq j$. Write a greedy algorithm that finds an Egyptian number representation for any given $r = \frac{m}{n}$. The input to your algorithm consists of integers n and m, and the output is list $[n_1, n_2, \dots, n_k]$. Prove that your algorithm works (and halts on every input!) and has worst-case time complexity $O(m)$. Assume that all arithmetic operations can be done in constant time.

4 Exercise 23.1-6, page 468. (Hint: notice that every element of the adjacency-matrix reveals one non-sink node. Namely, if $a_{ij} = 1$ then i is not a sink, and if $a_{ij} = 0$ then j is not a sink. By testing carefully selected $n-1$ elements of the matrix you should be able to elimate $n-1$ of the nodes. Then you can easily check if the only remaining candidate is a sink or not.)