Homework 5

Try to be precise and to the point. Your answers should be short.

1. Suppose $\mathcal{S} = \{a, b, c, d, e, g, i, j\}$, where the frequencies are

$$f_a = 0.25, f_b = 0.15, f_c = 0.16, f_d = 0.14, f_e = 0.15,\ f_g = 0.05, f_i = 0.04, f_j = 0.06.$$

Build an optimal prefix code for this date.

2. Consider the recurrence relation $T(n) \leq 2T(n/2) + Cn$, where C is a constant. This is derived in analysis of the running time of merge-sort or counting inversions problem. When you unroll this recurrence relation, you arrive at the following pattern:

$$T(n) \leq 2^k T(n/2^k) + kCn,$$

where $k = 1, 2, \ldots, \log(n)$. By plugging $k = \log(n)$ into the pattern, you get

$$T(n) \leq n \cdot T(1) + \log(n)Cn = O(n \cdot \log(n)).$$

For each of the following recurrence relations find the general pattern formula and find the running time $T(n)$ of the algorithm given by the recurrence relations:
(a) $T(n) \leq 3T(n/2) + Cn$.
(b) $T(n) \leq 2T(n/3) + C$.
(c) $T(n) \leq 5T(n/4) + Cn$.
(d) $T(n) \leq 2T(n/2) + C$.
(e) $T(n) \leq 9T(n/3) + Cn^2$.
(f) $T(n) \leq 8T(n/2) + Cn^3$.

3. Say that an array A with n elements has a *majority element* if more than half of its entries are the same. The elements of the arrays are not necessarily comparable (that is you cant ask the questions of the type $A(i) \leq A(j)$?). However, assume that you can ask the questions of the type $A(i) = A(j)$ in constant time. Design an algorithm that tells us if A has a majority element. Your algorithm should run in $O(n \cdot \log(n))$ time.

4. Solve Exercises 1 and 2 in Lecture Note 14.