CS 466/666
Assignment 2
Due: Noon Thursday October 9, 2008

1) [25 marks] Suppose you are given an arbitrary simple (no loops or multiple edges) directed graph.
   a) Give an $o(n^3)$ (yes that is a "little o") algorithm to find which nodes are in simple cycles (no node is repeated in one cycle) of length exactly 4 in this graph. You do not have to find the cycles, just report the nodes that are on at least one such cycle. So if (a,b,c,d,a) were one cycle and (a,c,e,f,a) another; your answer would include [a,b,c,d,e,f]
   b) Actually reporting all the cycles of length 4 in $o(n^3)$ time may be a problem; why?
   c) Give an $o(n^3)$ algorithm to find the length of the shortest cycle in such a graph *(Hint: the shortest cycle in such a graph will be simple. However, determining the existence of longer cycles (which are not necessarily simple) may help.)*

2) [20 marks] We have discussed optimal binary search trees where the keys have an ordering that must be respected by the tree. Indeed, given a sequence of probabilities we discussed an $O(n^2)$ method of finding the optimal tree. It turns out there is an $O(n \lg n)$ method to find the optimal tree if the all the probabilities of searching for an element in the set are 0, and we are just given the probabilities of elements in the "gaps" (leaves). You have also seen a linear technique for finding a Huffman code: that is an optimal binary tree for the case in which you can reorder the keys and all weights are associated with leaves. This question deals with the "4th" case which we will actually break into two:
   a) Given a set of probabilities of access of keys in sorted order, $(p_1 < \ldots < p_n)$ give an efficient algorithm to find the optimal binary tree when you may reorder the keys. What is the runtime of your method? Justify your answer.
   b) Assume we have a set of internal keys and a set of leaf keys and we require the leaf keys go to actual leaves of the tree and internal keys go to actual internal nodes of the tree. Given a set of probabilities of access of internal keys in sorted order $(p_1 < \ldots < p_n)$ and also the set of probabilities of access of leaf keys in sorted order $(q_0 < \ldots < q_n)$, give an efficient algorithm to determine the optimal tree when you may reorder the leaf keys and also the internal keys. What is the runtime of your method? Justify your answer.

3) [10 Marks] Consider the following simple method of finding the maximum of n distinct elements:

```
max ← A[0]
for i ← 1 to n-1 do  if max<A[i] then max ← A[i]
```

If the array A is increasing order, max is assigned n times; if A is in decreasing it is assigned just one. Suppose A is equally likely to be in any order; what is the expected number of times max is assigned. Give the exact number and also a very good approximation. Justify your answers.