# Problem Set 7 (due Tuesday, December 5)

## 1. (12 points) Reconstructing a tree

You are asked to reconstruct the reporting hierarchy of a huge company Disorganized, Inc., based on information that is complete but poorly organized. The information is available in an $n$-element array, in which each element of the array is a pair $(emp, boss)$ where $emp$ is the name of an employee and $boss$ is the supervisor of the employee, and $n$ is the number of employees in Disorganized. You may assume that the names of all employees are distinct. For the company CEO, the $boss$ entry is empty.

Your task is to compute a tree in which each node has the name of an employee $emp$ and a parent field pointing to the node corresponding to the supervisor of $emp$ (for the CEO, the parent field will point to NIL).

- **(a)** Design an $O(n \log n)$ time deterministic algorithm for the problem. Justify the running time of your algorithm.
- **(b)** Using hashing, design an expected $O(n)$ time randomized algorithm for the problem. Justify the running time of your algorithm.

## 2. (12 points) Ternary Huffman coding

The Huffman code algorithm that we studied in class encoded each character into sequences over the binary alphabet $\{0, 1\}$. Suppose you have a communication mechanism with which you can transmit sequences over a ternary alphabet $\{0, 1, 2\}$. Provide a modified Huffman algorithm for compressing a file with $n$ distinct characters with frequences $f_1, f_2, \ldots, f_n$ using $\{0, 1, 2\}$. Your algorithm should encode each character with a variable-length codeword over the values $0, 1, 2$ such that no codeword is a prefix of another codeword and so as to obtain the maximum compression. Prove that your algorithm is correct.

## 3. (12 points) Huffman encoding, entropy, and the English alphabet

- **(a)** Obtain a table from the web giving the frequencies of the letters of the English alphabet.
- **(b)** What is the optimum Huffman encoding of this alphabet?
- **(c)** What is the expected number of bits per letter?

*Entropy* is a mathematical formulation of the uncertainty and/or the amount of information in a data set. Consider a data set $D$ consisting of $n$ characters, each character independently chosen from a set $C$ according to a specified probability distribution $p$. That is, for $c \in C$ and $0 \leq i < n$, the probability that the $i$th character of $D$ is $c$ is $p(c)$. Note that $\sum_{c \in C} p(c) = 1$. The entropy of data set $D$ is then defined to be

$$n \sum_{c \in C} p(c) \log_2(1/p(c)).$$

Intuitively, the entropy measures the *information-theoretic minimum* number of bits needed to represent the data set.

**(d)** Calculate the entropy of a corpus, that contains English letters in the same frequencies as (a). Do you think this is the limit of how much English text can be compressed? Explain.

**(e)** **(6 bonus points)** Prove that if all the probabilities are powers of 2 (i.e., for every $c$ there exists an $i \geq 0$ such that $p(c) = 1/2^i$), then the expected number of bits used in the Huffman encoding of $D$ exactly equals its entropy.

### 4. (12 points) Security of a simplified RSA

Suppose that instead of using $N = pq$ in the RSA cryptosystem, we simply use a prime modulus $p$. As in RSA, we would have an encryption exponent $e$ relatively prime to $p - 1$ and encryption of a message $m \bmod p$ would be $m^e \bmod p$. Prove that this new cryptosystem is not secure, by giving an efficient algorithm to decrypt: that is, an algorithm that given $p$, $e$, and $m^e \bmod p$ as input, computes $m \bmod p$. Justify the correctness of your algorithm and analyze the running time of your decryption algorithm.

### 5. (12 points) RSA and digital signatures

A digital signature scheme has two components, `sign` and *verify*. Digital signatures can be implemented using RSA. The `sign` procedure takes a message $M$ and a secret key $d$ corresponding to an RSA triple $(N, d, e)$, then outputs a signature $\sigma = M^d \bmod N$. The *verify* procedure takes a public key $(N, e)$, a signature $\sigma$, and a message $M$, then returns "true" if $\sigma$ could have been created using a consistent RSA triple.

**(a)** Signing involves decryption, and is therefore risky. Show that if Bob agrees to sign anything he is asked to, Eve can take advantage of this and decrypt any message sent by Alice to Bob.

**(b)** Suppose that Bob is more careful and refuses to sign messages if their signatures look suspiciously like text. Describe a way in which Eve can nevertheless still decrypt messages from Alice to Bob, by getting Bob to sign messages whose signatures look random.