Homework 5

Fundamental Algorithms, Fall 2001, Professor Yap

- DUE: Mon Nov 26, in class.

1. [**10+20+10 POINTS**] Consider the following "Grouping Problem". Suppose we are given an input of the form (M, w) where
$$w = (w_1, w_2, , \ldots, w_n),$$
with M and the w_i's being non-negative numbers. A **solution** to this problem is any subdivision of the sequence w into some number of "groups", where each group has "size" at most M. Here a group G is a subsequence of **consecutive** numbers from w, of the form
$$G = (w_i, w_{i+1}, w_{i+2}, \ldots, w_j), \qquad 1 \le i \le j \le n.$$
The size of G is simply the sum $\sum_{\ell=i}^{j} w_\ell$. You are guaranteed that each $w_i \le M$ so that a solution exists. A subdivision of w into k groups can be represented by a sequence of k integers,
$$1 \le n_1 < n_2 < n_3 < \cdots < n_k \le n$$
where the ith group is given by $G_i = (w_{n_{i-1}+1}, w_{n_{i-1}+2}, \ldots, w_{n_i}\}$. Let us denote this subdivision as "$[n_1, n_2, \ldots, n_k]$". When $i = 1$ in this definition, we let n_0 be 0. You want a solution that is **optimal** in the sense that it has the smallest possible number of groups.

   Example. If $M = 5$ and $w = (1, 1, 1, 3, 2, 2, 1, 3, 1)$, one solution is the the subdivision $[n_1, n_2, n_3, n_4, n_5] = [2, 4, 6, 8, 9]$. This corresponds to the 5 groups $(1, 1), (1, 3), (2, 2), (1, 3), (1)$. But we can get 4 groups, using $[n_1, n_2, n_3, n_4] = [2, 4, 7, 9]$. This is also optimal.
   (i) Give a simple greedy algorithm for the Grouping Problem.
   (ii) Prove that your algorithm is optimal.
   (iii) Suppose the w_i's may be negative as well. Either prove that your algorithm is still optimal or show a counter example.

2. [**15 POINTS**] Compute the optimal Huffman code for the first two sentences of President Lincoln's Gettysburg address. The full address is reproduced below, but ONLY encode the first two sentences. You will need to encode spaces, commas and full stop, but NOT the newlines.

   In fact, define "white space" to be any maximal contiguous sequence of one or more space, newline and tab characters. The rule is to ignore the initial and terminal white spaces (if any), and to replace any remaining white space by a single space character.

   Note that small and capital letters are distinguished. You **must**
   (i) state the overall bit length of your coded string, and
   (ii) display the collection of intermediate code trees which are obtained in your computation.

3. [**10+15+10+10 POINTS**] Exercise 16.3-5, page 392. Representation of an optimal prefix code tree. However, instead of using $2n - 1$ bits to specify the tree, we allow you to use up to $4n - 4$ bits, so that the overall representation should use at most $4n - 4 + n \lceil \lg n \rceil$ bits. Note that the alphabet C is the binary representation of the numbers 0 to $n - 1$. You **must** do 3 things:
   (i) Prove that a Huffman tree with n leaves has exactly $2n - 2$ edges. (Use induction of the structure of the tree)
   (ii) Tell us how to construct the representation from any Huffman tree for the set $C = \{0, \ldots, n - 1\}$.
   (iii) Apply your code to the tree in Figure 16.4(b), assuming that $a = 0, b = 1, c = 2, \ldots, f = 5$.
   (iv) Describe how to reconstruct the Huffman code tree from your representation.

   MORE HINTS: You need to exploit the structure of the set C. How many edges does a Huffman tree with n leaves have?

4. [**20 POINTS**] We generalize the example of incrementing binary counters. Suppose we have a collection of binary counters, all initialized to 0. We want to perform a sequence of operations. Each operation is one of two types:

$$\texttt{inc}(C), \quad \texttt{add}(C, C')$$

where C, C' are names of counters. The operation $\texttt{inc}(C)$ increments the counter C by 1 as before. The operation $\texttt{add}(C, C')$ adds the contents of C' to C while simultaneously set the counter C' to zero. Assume that the cost to perform $\texttt{add}(C, C')$ is equal to the sum of the lengths of the binary numbers stored in C and C'. Show that this problem has an amortized cost that is constant per operation. HINT: the potential of a counter C should take into account the number of 1's as well as the bit-length of the counter.

5. [**15+15+15 POINTS**] Problem 17-2, page 426. Dynamic binary search. Our T.A. Sean had discussed this problem in recitation.

```
GETTYSBURG ADDRESS (for Problem 2):

Four score and seven years ago our fathers brought forth on this
continent a new nation, conceived in liberty and dedicated to the
proposition that all men are created equal.  Now we are engaged in
a great civil war, testing whether that nation or any nation so
conceived and so dedicated can long endure. We are met on a great
battlefield of that war. We have come to dedicate a portion of
that field as a final resting-place for those who here gave their
lives that that nation might live. It is altogether fitting and
proper that we should do this. But in a larger sense, we cannot
dedicate, we cannot consecrate, we cannot hallow this ground.
The brave men, living and dead who struggled here have consecrated
it far above our poor power to add or detract. The world will
little note nor long remember what we say here, but it can never
forget what they did here.  It is for us the living rather to be
dedicated here to the unfinished work which they who fought here
have thus far so nobly advanced. It is rather for us to be here
dedicated to the great task remaining before us--that from these
honored dead we take increased devotion to that cause for which
they gave the last full measure of devotion--that we here highly
resolve that these dead shall not have died in vain, that this
nation under God shall have a new birth of freedom, and that
government of the people, by the people, for the people shall
not perish from the earth.

-- Gettysburg, Pennsylvania.  November 19, 1863
```

The following exercises are NOT to be handed in, but we encourage you to try to solve them.

1. Problem 16.3-6, p.392. Generalize Huffman code to ternary codewords.
2. Give an algorithm for the Grouping problem when the numbers may be negative.
3. Problem 17.4-3. Question on Dynamic Tables.