Due at the start of class Th, May 6, 2004.

Problem 1. Assume that A_1, A_2, A_3, A_4 are doing to split a cake using the triming protocol. A_1 will make the first cut. Give a scenario where A_1 cuts a piece that is bigger than 1/4 (in his eyes) and ends up with less than 1/4 when the protocol is done.

Problem 2. Fifteen plates are placed evenly around a circular table, with name cards for fifteen guests. The guests fail to notice the cards until after they've sat down, at which point they discover that no one is in the correct seat. Show that the table can be rotated so that at least two of the guests are simultaneously correctly seated.

Problem 3. Given a knapsack of capacity K. Also given are n items, with weights w_i and profit p_i each (for item i). Show how dynamic programming can be used to find the maximum profit subset of items that fit in the knapsack (total weight of this subset should not exceed K).

Problem 4. Use the method described in class to obtain an optimal alignment for the strings $ACATT$ and $AATT$. Show the entire table, not just the final solution. Also show how to derive the optimal alignment *from* the table entries.

Problem 5. Show a Huffman tree encoding for the following example. The frequency of the following letters are:

letter	*frequency*
a	0.10
b	0.12
c	0.15
d	0.13
e	0.35
f	0.07
g	0.08

If we have a piece of text of 100,000 characters with this frequency distribution, what is the number of bits required to compress this piece of text.