Assignment 5, CS 251, March 16, 2019.

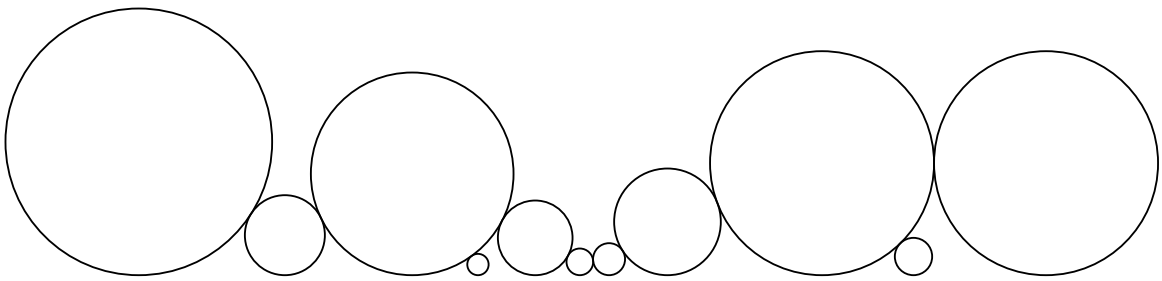
Exercise 1. LEMPEL-ZIV COMPRESSION. Write the algorithm for parsing a binary input sequence to prepare it for Lempel-Ziv compression. Assume that the input can be exactly parsed, i.e., that there isn't an incomplete last piece. The output is a sequence of pairs (i, j), where i refers to a piece number, and $j \in \{0, 1\}$. For example, the sequence 0 00 001 1 000 0011 10 0000 is parsed into (0,0), (1,0), (2,1), (0,1), (2,0), (3,1), (4,0), (5,0). Your algorithm should take time $O(n)$.

Exercise 2. CODING. How would you store a sequence of n 11-digit decimal numbers? Please discuss the optimality or near-optimality of your solution.

Exercise 3. HUFFMAN TREES. We define a Huffman tree on an alphabet A of size n, where for symbol i, p_i represents a frequency count (a positive integer). Someone has constructed a canonical Huffman tree (of size $2n-1$) for this, with the root having number 1. We have a table T of size n, where $T[i]$ has an integer-valued pointer to the leaf node in the Huffman tree for symbol i. Each of the $2n-1$ cells in the Huffman tree consists of left, right and parent pointers, a symbol number (which is only useful for leaves), a weight (i.e., a frequency or sum of frequencies), and next and previous pointers in a linked list that has the nodes in level order. It is also known that the weights in this level order are nonincreasing (one can always maintain this).

In an adaptive Huffman tree context, one would like to adjust this Huffman tree in $O(n)$ time if symbol i increases its frequency by one. (This operation can be iterated if we have a sequence of such increases.) Please write an algorithm for this.

Exercise 4. GREEDY METHOD IN ALGORITHM DESIGN: THE CIRCLE-PACKING PROBLEM. In the one-dimensional circle packing problem, one is given n radii $r(1), r(2), ..., r(n)$. Circles with these radii are packed in a box such that each circle touches the bottom of the box. The circles are arranged in the original order. The problem is to find the width of the minimum-sized box. The figure below illustrates what a solution might look like.



Design an efficient algorithm in terms of worst-case time as a function of n. Hint: for two adjacent touching circles with radii a and b, derive a formula for the distance between their centers. See what a solution looks like, and develop a greedy algorithm for it.