CPSC 320 Problem Set 1 18 September 2009
W. Evans Due: 25 September 2009 at the start of class.

1. A *shim* is a piece of metal or plastic that has a precise thickness. Prove that each integer thickness from 0 to $2^k - 1$ can be obtained by stacking a subset of the shims of thickness $2^0, 2^1, 2^2, \ldots, 2^{k-1}$.

2. Each of the n elements in an array A has value RED, WHITE, or BLUE. (It may be that the array contains no elements of a particular color.) Describe an algorithm that sorts the array A in increasing order (where RED < WHITE < BLUE) and takes time $O(n)$. The only operations you can perform are to examine an element to determine its color, and swap two elements in A (given their indices). (For example, you are not allowed to count the number of RED, WHITE, or BLUE elements.) Please explain the idea behind your algorithm, the loop invariant that it maintains, and why it takes $O(n)$ time.

   Hint: Try executing something like selection sort twice.

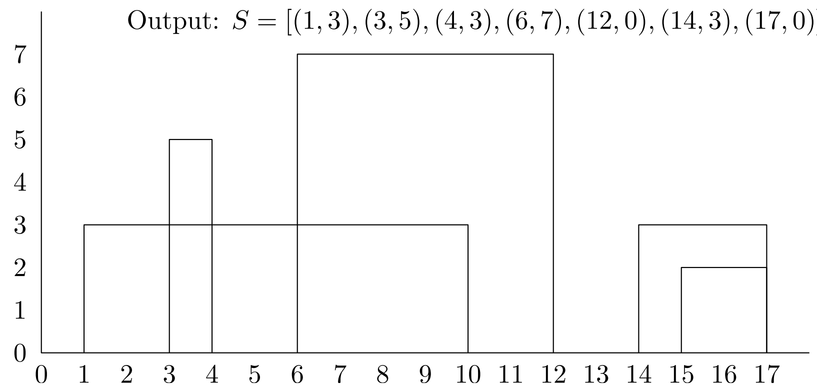
3. Let
$$f(n) = 2^{\lg(n)/2} n \lg(n/4) + 4^{\lg(n)} \log_5(n^{1.3})$$
   where $\lg = \log_2$. Find the slowest growing, simple function $g(n)$ such that $f(n) \in O(g(n))$.

4. The **Skyline Problem** is: Given an array, B, of n two-dimensional buildings, find the perimeter of the union of these buildings (the skyline). The ith building is specified by three real numbers, (L_i, H_i, R_i), which specify the left side x-coordinate, height, and right side x-coordinate of the building. Note that $L_i < R_i$. A skyline is an array $S = [(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)]$ of m pairs of real numbers where $x_1 < x_2 < \cdots < x_m$ and $y_m = 0$. For x-coordinates less than x_1, the skyline height is 0. Between x-coordinates x_i and x_{i+1}, the skyline height is y_i $(1 \leq i \leq m-1)$. For x-coordinates larger than x_m, the skyline height is $y_m = 0$.

   For example,

   Input: $B = [(14, 3, 17), (1, 3, 10), (3, 5, 4), (15, 2, 17), (6, 7, 12)]$

   Output: $S = [(1, 3), (3, 5), (4, 3), (6, 7), (12, 0), (14, 3), (17, 0)]$



   In class, we described a recursive divide-and-conquer algorithm that solves the Skyline Problem in time $O(n \lg n)$ assuming that we could merge two skylines S_1 and S_2 in time $O(|S_1| + |S_2|)$ (i.e., in time that is linear in their size). Describe such a merge algorithm.

5. List the following functions in order of increasing growth rate, i.e. if $f(n)$ precedes $g(n)$ in your list then $f(n) \in O(g(n))$. (In fact, if $f(n)$ precedes $g(n)$ then $f(n) \in o(g(n))$ will be true.)

$$n^3 \log n, \quad n^{\log_2 n}, \quad (\log n)^5, \quad 2^{(2^n)}, \quad \sqrt{n}, \quad n!, \quad n^2(\log n)^4, \quad 2^n, \quad \sum_{i=1}^{n} i^2$$

6. Solve each of the following recurrences exactly. Prove that your answers are correct. You may assume that n is a power of 2 in parts (b) and (c).

   (a) $T(n) = T(n-2) + 3$, $T(1) = 0$, $T(0) = 0$.

   (b) $T(n) = T(n/2) + 1$, $T(1) = 0$.

   (c) $T(n) = 4T(n/2) + n$, $T(1) = 0$.