CSC 5451, Professor E. Gethner
Assignment 2
17 September 2015
Quiz 2 takes place in class on Thursday, 1 October 2015

Please feel free to collaborate with one another on this assignment. Consider writing up the solutions on your own for quiz practice. **Important information regarding the quizzes: Be neat, write complete sentences, and show all of your work. The way you communicate the solution to your answer is as important as the answer itself.** Good luck!

1. Build the Huffman tree for the set of characters in this question. Include all characters including punctuation and spaces. How many bits are saved in the storage of this question using Huffman trees versus a storage based on a fixed-length encoding such as ASCII?

2. **(Closed Unit Intervals Covering Points)** CLR problem **16.2-5**: Describe an efficient algorithm that, given a set $\{x_1, x_2, \ldots, x_n\}$ of n points on the real line, determines the smallest set of closed unit-length intervals that contains all of the given points. Prove that your algorithm is correct. Note that the word "efficient" means that the complexity is no worse than polynomial-time. Be sure that your algorithm is clearly written in readable and understandable pseudocode.

3. **(Knapsack Practice)** A knapsack of volume K is to be (exactly) filled with a subset of n items, where k_j denotes the size of the j-th item.

   - Let $P(n, K)$ denote the original problem as stated in class.
   - The notation $P(i, k)$ denotes the knapsack problem for a knapsack of size $k \leq K$ with the first i items. That is, $P(i, k).exist =$**true** if the problem $P(i, k)$ has a solution with a subset of the first i items; if the i-th item is packed in the solution of $P(i, k)$, then $P(i, k).belong =$**true**.

   (a) Fill in the 2-dimensional array associated with the standard dynamic programming solution to the Knapsack problem with $K = 9$; entry (i, j) will contain a **0** if the *.exist* field is true but item i is *not* packed, will contain an **I** if the *.exist* field is true **and** item i has been packed, and will contain a "-" if there is no solution. There are four items to choose from, with sizes given by $k_1 = 1$, $k_2 = 2$, $k_3 = 4$, and $k_4 = 5$. I have already filled in part of the table as you can see.

   (b) Does $P(4, 9)$ have a solution? If yes, show how to use the matrix to decide exactly which items have been packed.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
$k_0 = 0$	**0**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
$k_1 = 1$	**0**	**I**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
$k_2 = 2$	**0**	**0**								
$k_3 = 4$	**0**	**0**								
$k_4 = 5$	**0**	**0**								

4. **(L-Tiling)** Consider C_n, a 2^n by 2^n checkerboard with the upper right square removed. An *L-tiling* of C_n is a tiling of C_n with L-shaped tiles (composed of three squares) with no overlaps and no square of the checkerboard left uncovered. Prove or disprove: for every $n \in \mathbb{Z}^+$, there is an L-tiling of C_n. See Figure 1 for pictures of C_1, C_2, and C_3, and Figure 2 for pictures of all possible L-shaped tiles.

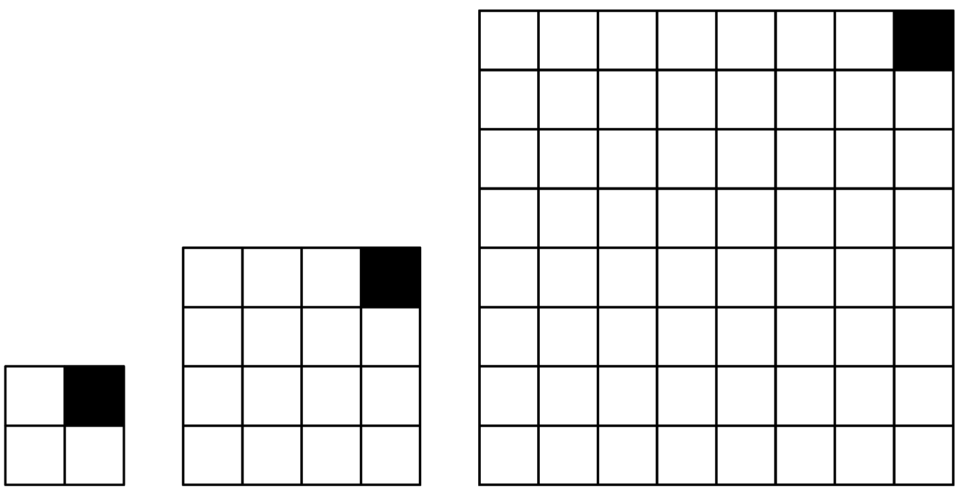

Figure 1: C_1**,** C_2**, and** C_3**: the black square has been removed from the board.**

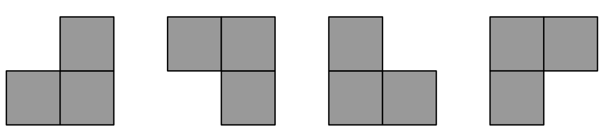

Figure 2: **All possible L-tiles composed of three squares.**