CS2302 - Data Structures

Spring 2023

Practice Exam 1, Programming Part

1. (General python) Write the function *divisible(n,a,b)* that receives three integers n, a, and b and returns 0 if n is not divisible by either a or b, 1 if n is divisible by either a or b (but not both), and 2 if n is divisible by both a and b.

2. (Lists) Write the function *sum_lists(L1,L2)* that receives two lists of integers $L1$ and $L2$, not necessarily of the same length, and returns a list of the same length as the longer of $L1$ and $L2$ where the first element is $L1[0] + L2[0]$, the second element is $L1[1] + L2[1]$, and so on. For example, *sum_lists([3,5,1],[2,8,9,1,2])* should return *sum_lists([5,13,10,1,2])*.

3. (Arrays) Write the function *reverse_odd_rows(A)* that receives a 2D array A and reverses all odd rows in A (that is, rows 1,3,5, etc.).

4. (Sets) Write the function *appears_in_all(L)* that receives a list of sets L and returns a set containing the elements that appear in ALL of the sets in L.

5. (Dictionaries) A sparse array is an array where most elements are zero. We can often reduce memory requirements by storing the positions and values of all non-zero elements of a sparse array. Write the function *sparse_array(A)* that receives a 2D array of integers A and returns a dictionary D containing the non-zero elements of A and their indices such that if $A[i, j] = k$ and $k > 0$, $D[(i, j)] = k$.

6. (Dictionaries) Write the function *longest_word(W)* that receives a list of words W and returns a dictionary D where $D[c]$ contains the longest word in W that starts with letter c, for every letter c in the alphabet. Break ties by choosing the word that appears first in W. For example, if W =*['alpha', 'beta', 'gamma', 'bear']*, $D['b']$ should be *'beta'*, since *'beta'* appears before *'bear'* in W.