Homework 3 Solution, Spring 2018

Task 1

Compile and run:

gcc -o instr instructor_client_sol.c list_hw_sol.c

./instr

Task 2:

In the notation below, N stands for the length of list A and M stands for the length of list pos_list.

moveAllMaxAtEnd(list A) – θ(N*N)

when all the max values are at the end and these max values are as many as half the list length, e.g. A is 1 -> 2 -> 1-> 2-> 5->5 -> 5-> 5. It will have to go N/2 nodes to get to each of the N/2 max values.

sublist(list A, list pos_list) – θ(N*M).

For each item in pos_list we have to iterate through all A to look for that position (using the nodeAtPos function which is O(N)).

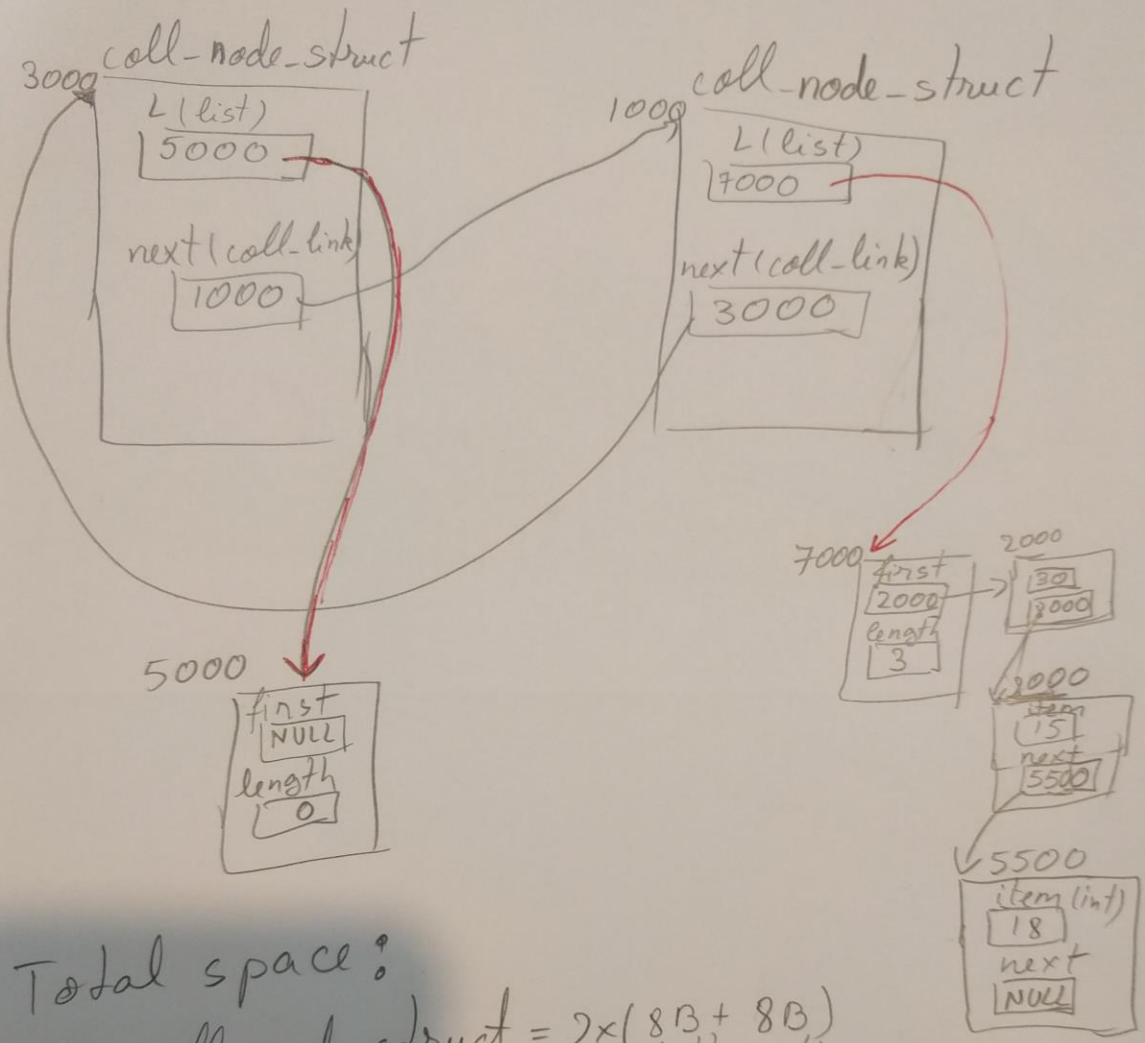
deleteOccurrences(list A, Item V) – θ(N)

It iterates through all the list A to compare the item in each node with the given value.

swapFirstThird(list A) – θ(1)

Does not depend on N.

Task 3:



Total space:

2 × coll_node_struct = 2 × (8B + 8B) (list) (next)

= 2 × 16B = 32B

empty list struct: 8B + 4B = 12B (first) (length)

list with 3 nodes: 12B + 3 × (4B + 8B) = 12 + 3 × 12 = 48B (list struct) (item) (next)

=> Total: 32B + 12B + 48B = 92B