**Assignment 4**

**Level 1:  
A.**

**(a) Write a recursive method, sum(L), in pseudo-code to calculate the sum of the  
integers in the List L of integers. Use positions to traverse the list. See the hint in  
the lecture notes. Analyze line by line your algorithm.**

**Algorithm sum(L)**

if L.isEmpty() then 1

return 0 1

else

return sumHelper(L,L.first()) n

Algorithm sumHelper(L,p)

if L.isLast(p) then 1

return p.element() 1

else

return p.element() + sumHelper(L, L.after(p)) n

**(b) Write a second recursive algorithm that uses the rank-based operations to  
traverse the list to calculate the sum of the integers in a Sequence. Analyze your  
algorithm line by line.**

**Algorithm sum(S)**

if S.isEmpty() then 1

return 0 1

else

return sumHelper(S,0) n

Algorithm sumHelper(S,r)

if r == S.size() then 1

return 0 1

else

return elemAtRank(r) + sumHelper(S, S..after(r)) n

**(c) Choose the better algorithm, either (a) or (b), then implement that algorithm in  
JavaScript using the List.js file or Sequence.js file provided in a previous  
assignment.**

The running time is O(n) for both the algorithms.

**B. Design a pseudo-code recursive method, findMax(L), that returns the maximum  
number in the list L. Implement in JavaScript using the List.js file provided  
previously.**

Algorithm findMax(L)

If L.isEmpty() then

return Error message

return FinaMaxhelper(L, L.first())

Algorithm FinaMaxhelper (L, p)

If L.size () = 0 then

return Error message

max := p.element()

If L.isLast(p) then

return p.element()

y := FinaMaxhelper (L, L.after(p))

if y > max then return y

else return max

**Level 2:  
C-2-4 Describe a pseudo-code recursive algorithm, setOfSubsets(n), that enumerates all  
of the subsets of the set of numbers {1,2,…,n}. What is the running time of your  
method?**