GETMAXASCENDSEQUENCE(InputSeq)

‘’’a list of integers, returns the subsequence that has the largest maximum.’’’

biggestSequence <- 0 (1)

subsequence <- [] (1)

sameSize <- False (1)

listOfStartingIndex <- [] (1)

number <- 0 (1)

i <- 0 (1)

startingIndex <- 0 (1)

While i ≠ length of InputSeq (n)

#always increases the number, for each number in list

Number <- number + 1 (n)

//don’t need to check if in ascending order as last number.

If I = length of InputSeq (n)

if number > biggestSequence (n)

//incase two or more subsequences of same size then a bigger //one, it makes it so it only prints the bigger one.

sameSize <- False (n)

//As a larger seqeunce found, it removes any starting //indexes //which were added. As otherwise it would print the previous same //sized subsequences if there is another one which is the same //size as the newest biggest size.

listOfStartingIndex <- [] (n)

// gets the starting index of that subsequence does this //through taking away the amount of numbers in the

//sequence from the total we have covered so far.

//as done immediately after the ascending order has been broken,

//taking away the number and then adding one will give you

//the starting index.

startingIndex <- i-number +1 (n)

//adds it to the list that will be used if there is multiple //subsequences of the same size so that we still print the first //subsequence.

listOfStartingIndex append startingIndex (n)

biggestSequence <-number (n)

//makes it 0 so that the number of consecutive numbers in //ascending order can be counted a fresh without the previous's //subsequence streak included.

number <- 0 (n)

else if number = biggestSequence (n)

//indicates whether their is multiple subsequences

//that are the same size and are the largest so far.

sameSize <- True (n)

listOfStartingIndex append (I – number+1) (n)

number <- 0 (n)

else (n)

number <- 0 (n)

else (n)

//if it is in fact not in ascending

If inputSeq[i+1] <= inputSeq[i] (n)

if number > biggestSequence (n)

sameSize <- False (n)

listOfStartingIndex <- [] (n)

startingIndex <- i-number +1 (n)

listOfStartingIndex append startingIndex (n)

biggestSequence <- number (n)

number <- 0 (n)

else if number = biggestSequence (n)

sameSize <- True (n)

listOfStartingIndex append (I – number+1) (n)

number <- 0 (n)

else (n)

number <- 0 (n)

//if their was more than one sequence that was the largest

if sameSize = True (1)

//adds every subsequence to a new list

for i to length of listOfStartingIndex (n)

append to subsequence (InputSeq[listOfStartingIndex[i]] to InputSeq[listOfStartingIndex[i]+number) (n)

return subsequence (1)

else (1)

//adds the highest subsequence to a new list

subsequence <- InputSeq[listOfStartingIndex[0]] to InputSeq[listOfStartingIndex[0] + number] (1)

return subsequence (1)

Runtime = 33n +12

Big O: O(n)