

HW 13-6.1

6.1) Linear time algorithm for max contiguous subsequence problem

Input: list of # a_1, a_2, \dots, a_n

Output: contiguous subsequence of max sum

ex) $\{5, 15, -20, 10, -5, 40, 10\} \rightarrow \{10, -5, 40, 10\} = 55$

maxSubSequence(array)

subsequence = $\{ \}$, curr_sum = 0

* For each array[j], there are 2

for all j in array.size - 1

choices - prev_values + array[j] extends

curr_value = max(array[j], array[j] + curr_sum) the sequence, or a new sequence
if curr_value < array[j] subsequence broke starts from array[j]

curr_sum = 0

clear result set: subsequence then append array[j]

else

curr_sum += curr_value

subsequence.append(array[j])

return subsequence

ex) $\{5, 15, -20, 10, -5, 40, 10\}$

array[0]

1) curr_val = max(5, 5+0) 2) array[1] curr_val = max(15, 15+5) 3) array[2]

sum = 5

sum = 20

curr_val = max(-20, 20-20)

sequence = $\{5\}$

sequence = $\{5, 15\}$

sum = -10 sequence = $\{5, 15, -20\}$

4) array[3]

5) array[4]

6) array[5]

7) array[6]

curr_val = max(10, -10+10)

curr_val = max(-5, 0-5)

curr_val = max(40, 5+40)

curr_val = max(45, 45+10)

sum = 10

sum = -5

sum = 45

sum = 55

sequence = $\{10\}$

sequence = $\{10, -5\}$

sequence = $\{10, -5, 40\}$

sequence = $\{10, -5, 40, 10\}$

Break

✓