QUIZ 2 (COL 351)

Name **Entry No.**

Give precise arguments. Needlessly long explanations will not fetch any marks.

You are given an array A of length n containing integers (which could be positive or negative). A sub-array $A[i, j]$ of A, where $i \leq j$, is defined by the sequence $A[i], A[i+1], \ldots, A[j]$. Give an $O(n \log n)$ time algorithm to find the sub-array of A with the largest sum. Justify your answer.

Example: Suppose A is $\{-8, 10, -6, 17, 2, -1\}$. Assuming that the first element of A is denoted by $A[1]$, the sub-array $A[1, 4]$ has total sum $-8 + 10 - 6 + 17 = 13$, whereas sub-array $A[3, 6]$ has total sum $-6 + 17 + 2 - 1 = 12$.

We use divide and conquer.

In the base case, when array size is 1, just output the array entry. Now assume that the array size is more than 1. Divide the array into two sub-arrays, $A_1 = A[1, n/2]$ and $A_2 = A[n/2 + 1, n]$ and solve the two sub-problems recursively . We get two solutions with sums s_1 and s_2, and let s be the maximum of s_1 and s_2 . Assume by induction (on the array size) that s_1 and s_2 are the maximum sub-array sum in the two arrays respectively. Then we see that s is the maximum sub-array sum for any sub-array which remains inside A_1 or A_2 only. We need to see if we can get a better sum by a sub-array which starts before (and including) $n/2$ and ends after $n/2$. This can be done as follows: for each $i \leq n/2$, let a_i be the sum of the elements in the sub-array $A[i, n/2]$ and for each $j > n/2$, let b_j be the sum of the elements in the sub-array $A[n/2 + 1, n]$. Clearly, the best such sub-array will have sum $\max_{i \leq n/2} a_i + \max_{j > n/2} b_j$. We output the maximum of this quantity and s . Clearly, the recurrence is $T(n) \leq 2T(n/2) + O(n)$, and so, running time is $O(n \log n)$.