

1) Let A be an array containing n numbers (positive and negative). Develop an algorithm that finds the two indices $1 \leq i \leq j \leq n$ such that $S_{ij} := \sum_{k=i}^j A[k]$ is maximized. For example, in the array $A = [10, -12, 5, 7, -2, 4, -11]$, the sub-array $A[3, 6]$ has the sum $S_{3,6} = 5 + 7 - 2 + 4 = 14$ and no other sub-array contains elements that sum to a value greater than 14, so for this input the algorithm should output $(3, 6)$. Write an efficient code for the above.

2) An array $A[1 \dots 2n + 1]$ is wiggly if $A[1] \leq A[2] \geq A[3] \leq A[4] \geq \dots \leq A[2n] \geq A[2n + 1]$. Given an unsorted array $B[1 \dots 2n + 1]$ of real numbers, write an efficient code that outputs a permutation $A[1 \dots 2n + 1]$ of B such that A is a wiggly array.