8 Algorithms 1 (fms27)

(*a*) An object has a method `gulp(x)`, which accepts a 64-bit floating point number. On its n-th invocation since the creation of the object, this method returns the median of the n numbers fed to the object up to that point. Can this method be implemented with worst-case asymptotic cost of $O(1)$ in both time and space (with respect to n)? Provide either clear pseudocode or a clear explanation of impossibility. [6 marks]

[*Note:* The median of $m > 0$ values is defined as follows. Imagine the m values have been sorted, and are indexed from 0 to $m - 1$. If m is odd, the median is the value at position $\frac{m-1}{2}$. Otherwise, it is the average of the two values closest to the midpoint, i.e. those at positions $\frac{m}{2} - 1$ and $\frac{m}{2}$.]

(*b*) Given an unsorted singly-linked list of n integers, each of which painted in one of the seven colours of the rainbow, we seek an algorithm of optimal worst-case asymptotic complexity for rearranging the list so that all the integers of the same colour are adjacent (forming a contiguous *region* of consecutive list elements with the same colour), and the coloured regions appear one after the other in the same order as in the rainbow, and the integers within each region retain the same relative order as they did in the original list.

(*i*) Give a clear, concise and complete description of your strategy. [6 marks]

(*ii*) Give clear pseudocode implementing your strategy from Part (*b*)(*i*). Details of pointer manipulation are required for full marks. [6 marks]

[*Note:* Please use the following notation: `l.head` = pointer to head node of list `l`; `n.colour` = colour of node `n`; `n.value` = integer value of node `n`; `n.next` = pointer to the successor of node `n` in the corresponding list; and so forth, defining and using new fields as appropriate.]

(*iii*) Derive the worst-case asymptotic complexity of your approach and prove it is optimal. [2 marks]