1. Dado um vetor A com n números inteiros, suponha que este vetor está "quase"ordenado em ordem crescente, sendo que talvez haja **no máximo** um elemento fora de ordem. Escreva um método de complexidade $\Theta(n)$ que dado o vetor A, descobre se há este elemento fora de ordem e, se houver, coloca este elemento na sua devida posição de modo que o vetor fique realmente ordenado. Justifique suas respostas.

2. Imagine a seguinte variação do algoritmo $MergeSort$: (i) a divisão é realizada em três subseqüencias, (ii) cada subseqüência é ordenada recursivamente e (iii) na fase de conquista, o 1^o terço é intercalado com o 2^o terço e o resultado é intercalado com o 3^o terço. Apresente a equação de recorrência desse algoritmo e apresente a complexidade desse algoritmo. Compare essa variação com o algoritmo $MergeSort$ tradicional. Justifique suas respostas.

3. Suponha que temos uma seqüência S de n elementos ($1 \leq n \leq 10000$), de forma que cada elemento em S representa um voto diferente para líder estudantil, dado como um inteiro m de 7 digitos ($1000000 \leq m \leq 9999999$) representando o número de matrícula do candidato escolhido. Planeje um algoritmo de complexidade de tempo $O(n \log n)$ (no caso médio) para determinar quem vence a votação representada por S, supondo que o candidato com o maior número de votos será o escolhido (mesmo se existir $O(n)$ candidatos). Suponha que a memória é escassa e portanto deve-se minimizar a quantidade extra (além da seqüência S) de memória a ser utilizada nesse algoritmo. (Retirado de [2] - página 471).

   Se não houvesse essa restrição de memória, você conseguiria implementar um algoritmo mais eficiente (por exemplo, $\Theta(n)$) ? Justifique suas respostas.

4. Os professores Howard, Fine e Howard propuseram o seguinte algoritmo de ordenação "elegante": (Retirado de [1] - página 130)

   STOOGE-SORT(A, i, j)

   if (A[i] > A[j]) **then** trocar A[i] $\leftrightarrow$ A[j]

   if (i + 1 $\geq$ j) **then return**

   k $\leftarrow \lfloor (j - i + 1)/3 \rfloor$

   STOOGE-SORT(A, i, j - k)

   STOOGE-SORT(A, i + k, j)

   STOOGE-SORT(A, i, j - k)

   - Qual é a idéia por trás desse algoritmo ? Como ele ordena o vetor A ?
   - Forneça uma equação de recorrência para o tempo de execução no pior caso de STOOGE-SORT e um limite assintótico restrito (notação $\Theta(n)$) sobre o tempo de execução no pior caso. Justifique suas respostas.