

Prova 2- Parte 1 (Vale 3 pontos)

1. Ilustre o fluxo de execução do algoritmo de ordenação Merge Sort, apresentando os processos de decomposição e composição de funções recursivas para o vetor $A = [2, 40, 51, 25, 37, 56, 8, 48]$ (1,0)
2. Reescreva o algoritmo de Intercalação (Merge), para um vetor $A[p, q, r]$ de modo que ele não utilize sentinelas e, em vez disso, pare tão logo todos os elementos do arranjo L ou do arranjo R tenham sido copiados de volta em A e então copie o restante do outro arranjo de volta em A . Implemente esse algoritmo em C ou em Python, em que o usuário entre com os valores do vetor A , de 10 posições, com $q-p+1=5$ e $r-q=5$. (1,0)
3. Para o algoritmo de busca binária, considere que o vetor A , seja inteiro e que esteja ordenado. Escreva o algoritmo Busca-Binaria-Recursiva($A, v, \text{inf}, \text{sup}$), em que v é o valor que se verifica se tem em A , inf a posição inferior do vetor e sup a posição superior do vetor. Se o valor de v estive contido em A retorne o índice do vetor A , em que v está contido. Se v não estiver em A retorne a palavra NIL. Determine o tempo de execução desse algoritmo no pior caso em notação Θ . Implemente esse algoritmo em C ou python, e teste em uma função principal em que o usuário entre com os valores de A e de v . (1,0).