

LISTA OBRIGATÓRIA PARCIAL DE PROJETO E ANÁLISE DE ALGORITMO

SEMANA 4

Para os problemas a seguir faça:

- a) Estruture a solução em pseudo-linguagem;
 - b) Calcule a complexidade do algoritmo para atestar que atende ao exigido;
 - c) Implemente o algoritmo e teste-o.
1. Dados dois conjuntos S e T , e um número real x , determine se existe um elemento de S e um de T cuja soma dê exatamente x . A complexidade do algoritmo deve ser $O(n \log n)$, onde n é o número total de elementos em ambos os conjuntos. Este é o problema 6.23, pag 177 de Udi Manber.
 2. O *número de inversões* de um vetor V de n elementos é o número de pares ordenados (i, j) tais que $1 \leq i < j \leq n$ e $V[i] > V[j]$. Escreva uma função que calcule o número de inversões de um vetor dado. O consumo de tempo de sua função deve ser $O(n \log n)$ no pior caso. Este é o Problema 2-4, pag 41 de Cormen et al (Inversões), similar ao problema 6252 do SPOJ.
 3. Considere um vetor com números inteiros positivos, nulos e negativos. Elabore um algoritmo com complexidade de tempo e espaço $O(n)$ para que no vetor resultante os elementos negativos precedam os elementos nulos e estes precedam os positivos.
 4. Considere a entrada um vetor com valores inteiros não negativos. Ordene estes valores segundo o seguinte critério: primeiro os pares e depois os ímpares. No entanto, os pares devem ser ordenados em ordem crescente e os ímpares em ordem decrescente. Você deve adaptar o Quicksort para resolver o problema (similar ao problema 1259 do URI)