

MC448 – Análise de Algoritmos
Lista de Exercícios 1

Orlando Lee

1. (CLRS 2.1-3) Considere o **problema da busca**:

Entrada: um vetor $A[1..n]$ e um valor x .

Saída: Um índice i tal que $A[i] = x$ ou -1 se x não aparece no vetor.

Escreva um pseudo-código para o algoritmo BUSCA LINEAR que varre o vetor da esquerda para a direita procurando pelo valor x . Escreva um invariante de laço que garante que o algoritmo está correto.

2. (CLRS 2.2-2) Considere o seguinte método para ordenar um vetor $A[1..n]$. Encontre o menor elemento em A e troque-o de posição com $A[1]$. Então procure o segundo menor elemento de A e troque-o de posição com $A[2]$ (note que para isto basta procurar o ??? elemento em $A[2..n]$). Repita este processo para os primeiros $n - 1$ elementos de A . Escreva um pseudo-código para este algoritmo que é conhecido como **algoritmo de seleção** (*selection sort*).

Escreva um invariante de laço que garante que o algoritmo está correto. Por que é suficiente considerar apenas os $n - 1$ primeiros elementos?

Faça uma análise de complexidade detalhada como feito em aula para o ORDENA-POR-INSERÇÃO (*Insertion sort*). Qual é o melhor e o pior caso?

3. (CLRS 2.3-5) Considere novamente o **problema da busca** supondo agora que o vetor $A[1..n]$ está **ordenado** em ordem crescente. A idéia consiste em tomar o elemento do “meio” $A[\lceil n/2 \rceil]$, comparar com x e eliminar uma das “metades” $A[1.. \lceil n/2 \rceil - 1]$ e $A[\lceil n/2 \rceil + 1..n]$. O método de **busca binária** consiste em repetir este processo eliminando metade do vetor a cada vez. Escreva um algoritmo **recursivo** que implemente esta idéia. Escreva uma recorrência que descreve a complexidade de tempo (no pior caso) e mostre que a solução é $\Theta(\lg n)$.