

Cota inferior para ordenação e ordenação em tempo linear

Ordenação em tempo linear

Questão 1. (CLRS) Exercícios: 8.1-1, 8.1-2, 8.2-1, 8.2-4, 8.3-1, 8.3-3, 8.4-1, 8.4-2,

Questão 2. (Manber) (6.21) A entrada é um conjunto S com n número reais. Projete um algoritmo de tempo $O(n)$ para encontrar um número que *não* está no conjunto. Mostre que $\Omega(n)$ é um limitante inferiores no número de passos para esse problema.

Questão 3. (Manber) (6.19) Dado um vetor de inteiros $A[1..n]$, tal que, para todo i , $1 \leq i < n$, temos $|A[i] - A[i+1]| \leq 1$. Seja $A[1] = x$ e $A[n] = y$, tais que $x < y$. Projete um algoritmo de busca eficiente para encontra j tal que $A[j] = z$ para um valor z , $x \leq z \leq y$. Qual é o número de comparações com Z que seu algoritmo faz.

Questão 4. (Manber) ** (6.20) Mostre usando árvore de decisão que o algoritmo que você desenvolveu no exercício anterior é ótimo no pior caso (ou melhore seu algoritmo até que você possa provar que ele é ótimo).

Questão 5. (Skiena) O Sr. B. C. Dull acredita ter desenvolvido uma nova estrutura de dados para filas de prioridade que suporta as operações de Inserção, Máximo e Extrair-Máximo, todas com tempo de execução $O(1)$ no pior caso. Prove que ele está enganado.

Questão 6. (Skiena) Suponha que o vetor $A[1..n]$ tenha apenas números de $\{1, \dots, n^2\}$, mas que no máximo $\log \log n$ desses números aparecem no vetor. Projete um algoritmo que ordene A em tempo substancialmente menor que $O(n \log n)$. Calcule a complexidade e demonstre a correção do algoritmo.