

Estruturas de Dados II - 2012-2

L4

Algoritmos de Ordenação e Complexidade de Algoritmos

Nas questões abaixo, você pode usar as seguintes funções:

qsort(v, N): ordena o vetor v de tamanho N em $O(N \cdot \log N)$

vmax(v, N): retorna a posição do maior elemento de um vetor de tamanho N em $O(N)$

vmin(v, N): retorna a posição do menor elemento de um vetor de tamanho N em $O(N)$

bsearch(v, N, num): retorna a posição do elemento **num** em um vetor de N elementos em $O(\log N)$.

Se não encontrar o elemento **num** no vetor, retorna -1.

- 1) Dado um vetor de inteiros de tamanho N , crie um algoritmo $O(N \cdot \log N)$ para informar qual número aparece mais vezes no vetor. Se dois ou mais números aparecem a maior quantidade de vezes, dê como resposta o maior desses inteiros.
- 2) Suponha que tenha descoberto um algoritmo mágico que consegue achar um pivô de um vetor em $O(1)$ de tal forma que $1/3$ dos elementos do vetor são menores ou iguais a esse pivô e outros $2/3$ são maiores. Prove que, com ajuda desse algoritmo mágico, você consegue implementar o *quicksort* em $O(N \cdot \log N)$.
- 3) Qual o melhor algoritmo de ordenação para usar nos casos a seguir e por quê?
 - a) Ordenar um vetor com as notas de alunos em uma turma, cada nota representada por um inteiro variando de 0 a 100.
 - b) Ordenar um vetor qualquer com 4 elementos.
 - c) Ordenar um vetor qualquer com 4000 elementos.
 - d) Ordenar as medidas de altura de um vetor contendo as alturas de todos os prédios de São Luís representadas por valores inteiros em metros.
- 4) Modifique o algoritmo de *insertsort* para encontrar o menor e o maior elemento do vetor em cada iteração e colocá-los na posição correta do vetor. Crie o código dessa versão alterada.