Lista de exercícios sobre pesquisa e ordenação
Prof. João B. Oliveira

Sobre pesquisa binária

1. Escreva um programa que cria um vetor `vet` (do tamanho que você preferir) e preenche o vetor com inteiros, já em ordem.

2. Escreva um método `pesqseq(int val)` que procura o valor `val` no vetor usando **pesquisa sequencial** e retorna a posição em que ele foi encontrado, ou -1 se não está no vetor.

3. Escreva um método `pesqbin(int val)` **não-recursivo**, que procura o valor `val` no vetor usando **pesquisa binária** e retorna a posição em que ele foi encontrado, ou -1 se não está no vetor.

4. Escreva um método `pesqbin(int val)` **recursivo**, que procura o valor `val` no vetor usando **pesquisa binária** e retorna a posição em que ele foi encontrado, ou -1 se não está no vetor.

5. Escreva um método `compare()` que procura por todos os elementos que estão no vetor e imprime o resultado da pesquisa binária e da sequencial. Estes dois resultados devem ser iguais para todos os elementos do vetor.

6. Escreva um método `mostrapesqs(int val)`que retorna o número de pesquisas no vetor usadas para encontrar (ou não) o valor `val`. Crie os métodos auxiliares que forem necessários e depois imprima uma tabela de todos os elementos do vetor com a quantidade de pesquisas usadas para encontrá-los.

Sobre quicksort

1. Escreva um programa que cria um vetor `vet` (do tamanho que você preferir) e preenche o vetor com inteiros, em ordem aleatória.

2. Escreva o método `quicksort(int []vet)` que ordena o vetor usando quicksort. Seu método deve ser *in-place*, ou seja, trabalhar sobre o próprio vetor, sem fazer cópias.

3. Escreva uma versão do quicksort que não é *in-place*, ou seja, pode operar com cópias de vetores.

4. Escreva o método `check(int []vet)` que verifica se o vetor está ordenado e use-o para testar o seu quicksort.