

Atividade Prática 1 (DUPLA)

Valor: 50% da 1ª Avaliação

Problema 1:

Elabore programa em Java que receba como entrada o vetor V1, que consiste em n números, bem como um vetor P que contenha valores p_0, \dots, p_k , que são números inteiros no intervalo $\{1, \dots, n\}$. Você pode assumir que $k < n$. O algoritmo deve gerar um vetor O que contenha o r_j -ésimo maiores n números, para cada $j \in \{1, \dots, k\}$. Portanto, se um $p_j = 5$ no vetor de entrada P, então queremos retornar o quinto maior elemento do vetor de entrada V1 como parte da saída.

Exemplo:

Entradas: V1 = [21, 19, 16, 13, 8, 5.3, 20]; P = [2, 5]

- Saída: [8, 19]

- Explicação: 8 é o segundo maior elemento de V1 e 19 é o 3 quinto maior elemento de V1.

[19, 8] também é um resultado aceitável.

Dica: estamos buscando um algoritmo com tempo de execução $O(n \log k)$.

Problema 2:

Elabore um programa em java que retorne a mediana de todos os elementos de dois vetores de comprimento n do mesmo tipo de dado. Considere que os vetores podem ser de números ou strings. Descreva a solução e descreva o comportamento assintótico do algoritmo proposto.

Problema 3:

Considere um vetor V de inteiros aleatórios com números entre 1 e K, em que K é um número inteiro. Queremos descobrir se existem dois elementos do vetor que estão a uma distância T um do outro, ou seja, que diferem em T unidades. Escreva um programa e java

- a) Escreva um programa em java simples que resolva o problema em $O(n^2)$
- b) Escreva um programa java simples que resolva em $O(n \log n)$.

Problema 4:

Implemente um programa em java denominado BHSI-Sort, combinando BuildHeap+ Selection Sort + Insertion Sort. Para usar esse algoritmo, o usuário deve inicialmente construir o Heap (Max ou Min). Em seguida, ordenar as extremidades definidas em %E do tamanho do vetor usando o SelecSort. Por último, ordenar os elementos centrais usando o InsertSort.

Experimentos:

- 1) Todos o métodos deverão implementar os objetos utilizando *Generics* (<https://docs.oracle.com/javase/tutorial/java/generics/types.html>).
- 2) Você deverá enviar, também, um relatório mostrando e discutindo os resultados obtidos. Mostre prints da execução e dos resultados.

3. Entrega

- Código fonte do programa em JAVA (bem indentado e comentado) utilizando os conceitos de Orientação a Objetos. Relatório com os resultados (Upload no SIGAA)

Referência:

<http://web.stanford.edu/class/cs161/Sections/Section3/Section3.pdf>