



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
Instituto de Ciências Exatas e Informática  
Trabalho prático sobre Métodos de Ordenação

Curso : *Engenharia de Software*  
Disciplina : *Algoritmos e Estruturas de Dados II*  
Professora : *Eveline Alonso Veloso*

### **Regras Básicas:**

1. Estude bastante cada par de entrada/saída fornecido.
2. Todos os programas deverão ser desenvolvidos na linguagem de programação Java.
3. Essas práticas poderão ser desenvolvidas em grupos de, no máximo, três integrantes.
4. Cópias de exercícios, se existirem, serão encaminhadas ao colegiado de coordenação didática do curso.
5. Para cada exercício, vocês devem submeter, no VERDE, apenas um arquivo (.java) por grupo, que contenha todo o código necessário à resolução do problema proposto.
6. A resolução (código) de cada exercício deverá ser submetida ao VERDE.
7. A execução do código submetido será realizada automaticamente pelo VERDE, mas o código será analisado e validado pela professora.

### **Base de Dados:**

[Airbnb](#) é um serviço *online* comunitário que possibilita às pessoas anunciarem, encontrarem e reservarem acomodações e meios de hospedagem.



Foi fundado em Março de 2008 por Brian Chesky, Joe Gebbia e Nathan Blecharczyk em São Francisco, na Califórnia. O financiamento inicial foi obtido a partir da incubadora **Y Combinator**. Mais tarde, Greylock Partners, Sequoia Capital e Ashton Kutcher também investiram na empresa.

A plataforma permite aos indivíduos alugarem toda ou parte de sua própria casa, como uma forma de acomodação extra. O *site* fornece um serviço de busca e reservas entre a pessoa que oferece a acomodação e o turista que busca pela locação. Abrange mais de 7,7 milhões de anúncios em mais de 100.000 cidades e 200 países.

No entanto, no mundo, centenas de cidades possuem restrições para locações de curto prazo. Na Europa, cidades como Barcelona, Amsterdã, Paris e Veneza, impuseram restrições à plataforma nesse sentido. Em Paris, por exemplo, os

anfitriões não podem alugar seus imóveis por mais de 120 dias por ano e estes devem estar regulares perante a prefeitura e adimplentes com os tributos. Nos Estados Unidos, cidades como Washington D.C., Los Angeles e Santa Monica, possuem restrições similares.

No Japão a situação não é diferente. A regulamentação nipônica exige registro dos anfitriões para que possam anunciar seus imóveis na plataforma. Ainda assim, limitando em 180 dias por ano o tempo que um imóvel pode ser alugado.

No Brasil ainda não há legislação que regulamente o Airbnb. No entanto, em abril de 2021, o Superior Tribunal de Justiça (STJ) decidiu que condomínios residenciais podem impedir o uso de imóveis para locação pela plataforma. Segundo a Quarta Turma do Tribunal, o sistema de reserva de imóveis pela plataforma digital é caracterizado como uma espécie de contrato atípico de hospedagem – distinto da locação por temporada e da hospedagem oferecida por empreendimentos hoteleiros, que possuem regulamentações próprias.

Fontes: **Airbnb:** <https://news.airbnb.com/br/about-us/> e <https://pt.wikipedia.org/wiki/Airbnb>  
Acessados em 27 de março de 2024.

## **Exercícios:**

### **1. Bolha**

Utilizando vetores, ordene registros de acomodações do Airbnb aplicando o algoritmo de ordenação **Bubblesort**, considerando que a chave de pesquisa seja o atributo **satisfação** (*overallSatisfaction*). Em caso de empate, o segundo critério de ordenação deve ser o atributo **id da acomodação** (*roomId*).

Utilize a classe *Acomodacao* especificada e desenvolvida em prática anterior.

Seu programa deve ler um arquivo-texto chamado **dados\_airbnb.txt** que, no VERDE, **localiza-se na pasta /tmp**. Você deve preencher um vetor de objetos da classe *Acomodacao* com os dados das diversas acomodações informadas nesse arquivo. A **primeira linha** desse arquivo deve ser **descartada**, pois apenas informa os diversos **campos** presentes em cada uma das linhas seguintes. Cada uma das linhas seguintes indica os dados de uma acomodação, separados por uma tabulação ('\t').

Seu programa também deve ler a entrada padrão que apresenta, em sua primeira linha, a **quantidade de acomodações** que devem ser **ordenadas**.

Em seguida, são informadas, ainda na entrada padrão, várias linhas sendo que cada uma contém o **id da acomodação** (*roomId*) cujos dados devem ser inseridos no vetor de acomodações a ser ordenado.

A saída padrão corresponde aos registros ordenados, um por linha. Em cada linha da saída, escreva os dados do registro correspondente obedecendo o seguinte formato:

[roomId ## hostId ## roomType ## country ## city ## neighbourhood ## reviews ## overallSatisfaction ## accommodates ## bedrooms ## price ## propertyType]

Além disso, crie um arquivo de *log* na pasta corrente com o nome **matrícula\_bolha.txt** com uma única linha contendo: seu número de matrícula, tempo de execução de seu algoritmo de ordenação (em milissegundos), número de comparações realizadas entre os elementos do vetor de acomodações e número de movimentações realizadas entre os elementos do vetor. Todas as informações desse arquivo de *log* devem ser separadas por uma tabulação '\t'.

## 2. Ordenação por inserção

Repita a questão de ordenação de acomodações do Airbnb por meio do método *Bubblesort*, contudo, aplicando o algoritmo de **ordenação por inserção**, considerando como chave de pesquisa o atributo que indica a **quantidade de pessoas que a acomodação suporta** (*accommodates*). Em caso de empate, o segundo critério de ordenação deve ser o atributo **id da acomodação** (*roomId*).

O nome do arquivo de *log* dessa questão será **matrícula\_insercao.txt**.

## 3. Ordenação por seleção

Repita a questão de ordenação de acomodações do Airbnb por meio do método *Bubblesort*, contudo, aplicando o algoritmo de **ordenação por seleção**, considerando que a chave de pesquisa seja o atributo **país** (*country*). Em caso de empate, o segundo critério de ordenação deve ser o atributo **cidade** (*city*) da acomodação. No caso de novo empate, o terceiro critério de ordenação deve ser o atributo **vizinhança** (*neighbourhood*) e, por fim, se necessário, o quarto critério de desempate deve ser o atributo **id da acomodação** (*roomId*).

O nome do arquivo de *log* dessa questão será **matrícula\_selecao.txt**.

## 4. Heapsort

Repita a questão de ordenação de acomodações do Airbnb por meio do método *Bubblesort*, contudo, aplicando o algoritmo de ordenação **Heapsort**, considerando que a chave de pesquisa seja o atributo **revisões** (*reviews*). Em caso de empate, o segundo critério de ordenação deve ser o atributo **id da acomodação** (*roomId*).

O nome do arquivo de *log* dessa questão será **matrícula\_heapsort.txt**.

## 5. Mergesort

Repita a questão de ordenação de acomodações do Airbnb por meio do método *Bubblesort*, contudo, aplicando o algoritmo de ordenação **Mergesort**, considerando que a chave de pesquisa seja o atributo **id do anfitrião** (*hostId*).

Em caso de empate, o segundo critério de ordenação deve ser o atributo ***id da acomodação*** (*roomId*).

O nome do arquivo de *log* dessa questão será **matrícula\_mergesort.txt**.

## **6. Quicksort**

Repita a questão de ordenação de acomodações do Airbnb por meio do método *Bubblesort*, contudo, aplicando o algoritmo de ordenação ***Quicksort***, considerando que a chave de pesquisa seja o atributo ***preço*** (*price*). Em caso de empate, o segundo critério de ordenação deve ser o atributo ***tipo da acomodação*** (*roomType*). No caso de novo empate, o terceiro critério de ordenação deve ser o atributo ***id da acomodação*** (*roomId*).

O nome do arquivo de *log* dessa questão será **matrícula\_quicksort.txt**.