

SCC0224 Estruturas de Dados II  
Prof. Thiago A. S. Pardo

### **Especificação do Trabalho Prático 1**

Com a crise que a sociedade enfrenta atualmente, a Computação tem sido uma ferramenta essencial! Tem sido utilizada para trabalhar em casa, para auxiliar pesquisas médicas, para rastrear e acompanhar a progressão da pandemia de COVID-19 e para propiciar entretenimento (streaming, por exemplo), entre diversos outros usos no dia a dia. É impensável como seria nossa rotina hoje se não tivéssemos o apoio da Computação.

Neste trabalho, vamos usar nosso conhecimento acumulado de algoritmos e estruturas de dados para desenvolver um sistema de apoio à tomada de decisões sobre políticas públicas no combate à pandemia.

Quando solicitado, o sistema deve ler as informações sobre a evolução da COVID-19 nas várias localidades do mundo a partir do arquivo csv disponibilizado publicamente em <https://www.kaggle.com/imdevskp/corona-virus-report> (use o arquivo nomeado covid\_19\_clean\_complete.csv, também disponibilizado no Tidias). Esse arquivo contém uma listagem de localidades, datas e situação da doença, com 8 colunas por item/registro: província/estado, país/região, latitude, longitude, data, casos confirmados da doença, mortes pela doença e casos de cura. Atenção: como verão, alguns registros estão incompletos. Com base nessas informações lidas e armazenadas em memória, o sistema deve realizar várias operações, como ordenar os dados por país/região, latitude ou longitude, recuperar os dados de um país/região por data, etc.

Seu sistema deve ter minimamente as seguintes funcionalidades disponíveis para o usuário:

1. (a) Ler o arquivo de dados e armazenar os registros em uma lista sequencial (fique à vontade para decidir por estática ou dinâmica) na memória para, em seguida, (b) ordená-los de forma crescente por país/região, latitude ou longitude (critério que deve ser informado pelo usuário) e data (ou seja, não basta ordenar por localidade, é preciso, para cada localidade, fazer uma segunda ordenação por data) e (c) imprimir a listagem ordenada em um arquivo de saída e o tempo de execução da ordenação na tela do usuário. Você deve implementar um método de ordenação simples de sua escolha para isso (bubble-sort, insertion-sort ou selection-sort).
2. (a) Ler o arquivo de dados e armazenar os registros em uma lista sequencial (fique à vontade para decidir por estática ou dinâmica) na memória para, em seguida, (b) ordená-los de forma crescente por país/região, latitude ou longitude (critério que deve ser informado pelo usuário) e data (ou seja, não basta ordenar por localidade, é preciso, para cada localidade, fazer uma segunda ordenação por data) e (c) imprimir a listagem ordenada em um arquivo de saída e o tempo de execução da ordenação na tela do usuário. Você deve implementar um método de ordenação avançado de sua escolha para isso (quick-sort, shell-sort ou heap-sort).

3. (a) Ler o arquivo de dados e armazenar os registros em uma tabela hash, cuja função hash deve receber como entrada o nome do país/região e uma data e, com base em hashing, (b) permitir que o usuário consulte (quantas vezes quiser) o número de casos, mortes e curas por país/região e data. Fique à vontade para decidir pela melhor estratégia de hashing (fechado ou aberto) e de tratamento de colisões.
4. (a) Ler o arquivo de dados e armazenar os registros na memória e, (b) com base em duas datas especificadas pelo usuário, processar os registros armazenados e informar o país/região que teve o maior número de mortes no período indicado, informando também esse número. Em caso de empate, liste todos os países/regiões empatados e seus números. Você deve decidir como fazer esse processamento, ou seja, como armazenar os dados, que métodos de ordenação e busca vai usar (e se vai usar), etc.

Você também pode implementar outras funcionalidades que possam ser úteis além das obrigatórias listadas acima. Isso pode incrementar sua nota de trabalho.

Lembre-se de usar o bom senso e de levar em consideração todo o aprendizado sobre tipos abstratos de dados, eficiência de algoritmos, melhores práticas de implementação, etc. Tudo isso será avaliado. Também é muito importante que seu sistema seja claro e fácil de usar. Coloque-se no lugar do possível usuário e avalie se você estaria satisfeito em usar um sistema como o seu.

O trabalho prático deverá ser entregue até o **dia 18 de maio**, podendo ser feito em **duplas**. A dupla deverá submeter em Atividade no Tidia um arquivo compactado com o seguinte (basta que um membro da dupla faça a submissão): o código-fonte completo do trabalho e um relatório. O relatório deverá conter os nomes dos membros do grupo, os detalhes e decisões de projeto para implementar cada um dos itens anteriores e um exemplo de execução do sistema. Após entregar, a dupla deve **agendar com os estagiários PAE uma data para apresentar o trabalho** (virtualmente).