

Universidade Federal de Mato Grosso
Campus Universitário do Araguaia
Instituto de Ciências Exatas e da Terra
Curso: Ciência da Computação

Estruturas de dados I
Análise dos dados de Covid por meio da
estrutura Heap

Docente: Prof. Dr. Ivairton Monteiro Santos
Discente: Braz Amorim Campos

Barra do Garças - MT
Abril de 2025

Introdução

Este relatório busca descrever o processo de extração e ordenação de dados de um dataset sobre Covid-19 usando a estrutura Heap e uma lista encadeada simples.

Desenvolvimento

1 - Etapas e organização do programa

O trabalho foi realizado seguindo diferentes etapas:

1.1 - Escolha de uma base de dados

O dataset organizado em um arquivo XLS escolhido foi o *COVID-19 Dataset*, o arquivo usado foi “country_wise_latest.xls”.

Disponível em: <https://www.kaggle.com/datasets/imdevskp/corona-virus-report?select=country_wise_latest.csv>

1.2 - Carregamento dos dados

Os dados foram extraídos e armazenados em uma lista dinâmica com encadeamento simples, a função escrita responsável por esse processo é a *main_carregaLista()*, que através das funções *fgets()* e *strtok()*, lê e extrai linha a linha do arquivo do dataset as informações, que são então inseridas em um nó auxiliar temporário que após ser preenchido, é usado pela função de inserção na lista *lista_inserInicio()* para inserir um novo nó na lista principal do programa.

1.3 - Implementação da estrutura heap e seu uso atrelado ao menu

Após todos os dados serem carregados com sucesso na lista dinâmica, o programa exibe ao usuário um menu onde ele pode escolher qual informação dentre três opções de dados já cadastrados nos nós da lista será usada como chave para a criação do heap máximo, quando o critério de seleção de chave é escolhido, o usuário então deverá informar a quantidade de dados a serem removidos do heap, dados estes que após sua remoção serão exibidos no terminal.

No caso do usuário escolher voltar ao menu e carregar o heap novamente, o contador da estrutura heap é zerado de forma que todos os dados da lista sejam novamente inseridos no heap, sem a permanência dos dados da remoção e visualização anterior.

2 - Análise dos resultados

Dessa forma, para demonstrar o funcionamento do programa, executando a retirada de 10 elementos da raiz do heap para cada um dos critérios, as seguintes saídas foram obtidas:

2.1 - Critério: Casos confirmados

[CRITERIO: confirmados] MOSTRANDO INFORMACOES REMOVIDAS DO HEAP

[AVISO] Removendo 10 elementos do heap

Pais/Regiao	Confirmados	Mortes	Recuperados	Mortes/100	Recuperados/100
US	4290259	148011	1325804	3.45	30.90
Brazil	2442375	87618	1846641	3.59	75.61
India	1480073	33408	951166	2.26	64.26
Russia	816680	13334	602249	1.63	73.74
South Africa	452529	7067	274925	1.56	60.75
Mexico	395489	44022	303810	11.13	76.82
Peru	389717	18418	272547	4.73	69.93
Chile	347923	9187	319954	2.64	91.96
United Kingdom	301708	45844	1437	15.19	0.48
Iran	293606	15912	255144	5.42	86.90

2.2 - Critério: Casos de morte

[CRITERIO: mortos] MOSTRANDO INFORMACOES REMOVIDAS DO HEAP

[AVISO] Removendo 10 elementos do heap

Pais/Regiao	Confirmados	Mortes	Recuperados	Mortes/100	Recuperados/100
US	4290259	148011	1325804	3.45	30.90
Brazil	2442375	87618	1846641	3.59	75.61
United Kingdom	301708	45844	1437	15.19	0.48
Mexico	395489	44022	303810	11.13	76.82
Italy	246286	35112	198593	14.26	80.64
India	1480073	33408	951166	2.26	64.26
France	220352	30212	81212	13.71	36.86
Spain	272421	28432	150376	10.44	55.20
Peru	389717	18418	272547	4.73	69.93
Iran	293606	15912	255144	5.42	86.90

2.2 - Critério: Casos de recuperação

[CRITERIO: recuperados] MOSTRANDO INFORMACOES REMOVIDAS DO HEAP

[AVISO] Removendo 10 elementos do heap

Pais/Regiao	Confirmados	Mortes	Recuperados	Mortes/100	Recuperados/100
Brazil	2442375	87618	1846641	3.59	75.61
US	4290259	148011	1325804	3.45	30.90
India	1480073	33408	951166	2.26	64.26
Russia	816680	13334	602249	1.63	73.74
Chile	347923	9187	319954	2.64	91.96
Mexico	395489	44022	303810	11.13	76.82
South Africa	452529	7067	274925	1.56	60.75
Peru	389717	18418	272547	4.73	69.93
Iran	293606	15912	255144	5.42	86.90
Pakistan	274289	5842	241026	2.13	87.87

Conclusão

A partir da remoção dos dados do heap máximo, pode-se concluir que devido ao comportamento da estrutura de sempre manter o maior valor na raiz (primeira posição do vetor estático), o método de ordenação heap sort (crescente) foi usado de forma “involuntária”, permitindo a visualização dos dados ordenados em ordem crescente de acordo com o critério de chave escolhido pelo usuário na medida em que foram sendo removidos.