

Projeto 2: Gerenciador de Concursos da Mega-Sena com Tabela de Dispersão

(versão 1.0)

Descrição

O projeto envolve a criação de um gerenciador de concursos da Mega-Sena que utiliza *uma tabela de dispersão* para armazenar e acessar os dados dos concursos. Além das operações básicas de inserção, busca e remoção, o sistema deve ser capaz de calcular e apresentar estatísticas sobre os resultados dos concursos.

Composição

Este projeto poderá ser implementado em dupla.

Objetivo

- Desenvolver um sistema para gerenciar concursos da Mega-Sena utilizando uma tabela de dispersão (*hash table*) para armazenar e acessar os resultados dos concursos.
- Implementar o sistema utilizando a linguagem C.
- Calcular e apresentar estatísticas relevantes sobre os números sorteados.
- **Usar a nota deste projeto para compor a nota da terceira avaliação.**

Funcionalidades

1. Inserir Concurso:

- Além do carregamento dos dados do histórico de concursos, o usuário pode inserir um novo concurso na tabela de dispersão. Cada concurso é identificado por um número único e contém a data do sorteio e os seis números sorteados.
- O sistema deve garantir que não haja duplicação de números de concurso.

2. Buscar Concurso:

- O usuário pode buscar um concurso específico na tabela de dispersão usando o número do concurso.
- A busca deve retornar a data do sorteio e os números sorteados.

3. Remover Concurso:

- O usuário pode remover um concurso da tabela de dispersão usando o número do concurso.
- A remoção deve garantir que o concurso seja excluído corretamente da tabela.

4. Visualizar Todos os Concursos:

- O sistema deve permitir a visualização de todos os concursos armazenados na tabela de dispersão.
- A visualização deve exibir o número do concurso, a data do sorteio e os números sorteados.

5. Carregar Concursos de um Arquivo:

- O sistema deve carregar concursos a partir de um arquivo de dados. O arquivo deve conter resultados históricos dos sorteios da Mega-Sena (no formato CSV ou TSV).

6. Apresentar Estatísticas:

- **Quantidade de sorteios de um número:** Mostrar a quantidade de sorteios de um número específico.
- **Dez Números Mais Sorteados:** Mostrar os dez números que foram sorteados com maior frequência.

- **Dez Números Menos Sorteados:** Mostrar os dez números que foram sorteados com menor frequência.
- **Quantidade de concursos em um determinado ano:** Mostrar a quantidade de concursos que foram realizados em um determinado ano, bem como os seus dados.

Formato do Arquivo de Dados

- **Formato CSV:** O arquivo deve ter uma linha por concurso, com o número do concurso, a data do sorteio e os 6 números sorteados, separados por vírgulas. Exemplo:

```
Concurso,Data,bola_1,bola_2,bola_3,bola_4,bola_5,bola_6
2756,01-08-2024,1,46,13,18,6,43
2755,30-07-2024,58,7,17,33,41,13
2754,27-07-2024,55,10,44,14,56,58
```

- **Formato TSV:** O arquivo deve ter uma linha por concurso, com o número do concurso, a data do sorteio e os 6 números sorteados, separados por tabulação. Exemplo:

```
Concurso Data bola_1 bola_2 bola_3 bola_4 bola_5 bola_6
2756 01-08-2024 1 46 13 18 6 43
2755 30-07-2024 58 7 17 33 41 13
2754 27-07-2024 55 10 44 14 56 58
```

Requisitos

1. **Tabela de Dispersão:** Utilizar uma tabela de dispersão para armazenar e gerenciar concursos.
2. **Tratamento de Colisões:** Implementar a estratégia de encadeamento (*chaining*) para tratamento de colisões, usando listas ligadas.
3. **Arquivo de Dados:** Utilizar um arquivo CSV ou TSV para armazenar um histórico dos concursos (já fornecido). O arquivo deve ser lido e os dados inseridos na tabela de dispersão.
4. **Estatísticas:** Calcular e apresentar as estatísticas exigidas com base nos dados armazenados.

Estrutura da Tabela de Dispersão

1. Definição da Estrutura de Dados:

- **Nó:** Estrutura para armazenar os dados de cada entrada (número do concurso, data do sorteio e números sorteados).
- **Tabela de Dispersão:** Estrutura para armazenar a tabela de dispersão, com uma lista de células onde cada célula é uma lista ligada.

2. Função de Dispersão:

Função para mapear o número do concurso para um índice na tabela de dispersão.

3. Tratamento de Colisões:

Implementar encadeamento (*chaining*) para lidar com colisões, onde múltiplos concursos podem ser armazenados na mesma posição da tabela.

Observações

1. Certifique-se de que a implementação da tabela de dispersão e das listas ligadas esteja correta e eficiente.
2. O sistema deve lidar adequadamente com tentativas de remoção de concursos que não existem.
3. Inclua uma interface de usuário que permita a interação fácil com as funcionalidades do sistema.
4. O código deve ser bem comentado e documentado para facilitar a compreensão e manutenção.
5. Modularize o projeto (`.c`, `.h` e `makefile`).