

Problema 2 - MI-ALGORITMOS

Sistema para o Censo Demográfico de 2020

Wesley Ramos dos Santos¹

¹Universidade Estadual de Feira de Santana (UEFS)
Av. Transnordestina, s/n – 44036-900 – Feira de Santana – BA – Brazil

wesleysrw@gmail.com

Resumo. *Este relatório apresenta o desenvolvimento do problema 2 ofertado pela disciplina do MI-Algoritmos. Ele tem como objetivo mostrar todo o processo para resolução do problema, bem como as dificuldades encontradas, métodos para solucioná-las e referências utilizadas.*

1. Introdução

O problema 2 ofertado pela disciplina Mi-algoritmos teve como objetivo a elaboração de um piloto para que a coleta de dados do censo demográfico do IBGE seja feita através de um sistema automatizado.

O software precisaria permitir importar as regiões do país, seus estados e os municípios, fazer o cadastro dos técnicos e validar sua matrícula, só então se a matrícula for válida o sistema exibiria as informações tais como, números de domicílios utilizados para a coleta, número de domicílios particulares que já estão pagos, quantos ainda estão pagando e alugados, quantos por cidade possuem banheiro e quantos não possuem, a forma mais comum de abastecimento de água por cidade. Além disso o piloto também deve ser capaz de exibir o percentual de domicílios por cidade que ainda não possuem energia elétrica, de moradores que participaram da entrevista por cor ou raça e a região com maior número de municípios pesquisados. É importante ressaltar que o software pegará todos esses dados a partir de arquivos de texto, fará toda a validação usando os mesmos e por fim deverá disponibilizar ao usuário a opção de mostrar todos os dados ou um de cada vez.

Ao analisar todos esses fatos surgem as seguintes questões:

1. **Como usar os arquivos de texto no python?**
2. **Qual a melhor estrutura de dados para este software?**
3. **Como validar as informações?**
4. **Como fazer os cálculos?**
5. **Como fazer para dar a possibilidade de o usuário escolher como será exibido os dados?**

2. Introdução

Para sanar todas as dúvidas sanadas na sessão anterior foi utilizado o livro de introdução a programação com python [Menezes 2019], as respostas encontradas foram as seguintes:

Sobre a utilização obrigatória dos arquivos de texto para a resolução do problema, o python contém a função “open()”, que permite a importação dos arquivos de texto bem como sua manipulação.

Foi discutido na seção PBL que a melhor estrutura de dados para a resolução do problema em questão era o dicionário por questão de desempenho e facilidade na hora da busca de dados porém para resolução do problema foi utilizado “matriz” que no python nada mais é do que listas encadeadas já que pareceu mais fácil de implementar mesmo que graças a isso aja um maior gasto de memória. É importante ressaltar que para transformar o arquivo em matriz foi usado o comando “readlines()” que transformava o arquivo em uma grande matriz, dentro de uma estrutura de repetição “for” o comando “split()”, que transforma cada linha em uma lista separando os itens a partir do parâmetro “;” que foi passado dentro do split, formando assim uma lista dentro da outra ou seja uma “matriz”. Dentro do mesmo “for” foi colocado também o comando “rstrip()” para eliminar as quebras de linha.

Para validar o técnico e as informações contidas no arquivo de texto bastava comparar os arquivos “regiões.txt” e “técnicos.txt” com o arquivo “exemploPesquisa.txt” usando as posições das listas encadeadas dentro de uma estrutura de repetição “for” e o comando “len()” passando como parâmetro a matriz para ele percorrer toda a posição que fosse requerida, onde a primeira posição era representada pelo contador do “for” e a segunda era a posição do dado que queríamos na segunda lista.

Para os cálculos foi seguido uma linha de raciocínio parecida com a validação usando um “for” para percorrer a posição que eu queria na lista para cada um dos itens que era preciso adicionando valor a uma variável que foi usada para ficar contando a frequência absoluta. Para a porcentagem só foi necessário usar uma regra de três, levando em consideração a quantidade de repetições e a quantidade total daquele tipo de dado.

A interação com o usuário se dá através de um menu bem intuitivo, esse menu funciona através de várias estruturas de condição dentro de um loop infinito feito usando a estrutura de repetição “while”.

3. Conclusão

Mesmo com alguns percalços no caminho o problema 2 foi concluído com sucesso, não da melhor forma, pois em questão de performance teria sido melhor fazê-lo com dicionário porém ele funciona muito bem pelo que deu para perceber durante os vários testes.

Nele foram adicionados vários tratamentos erro então provavelmente não importa o que o usuário faça ele provavelmente não travara e fechara sozinho.

Um fator que causa uma certa insatisfação e que não deu para resolver por questões de erros técnicos, que ocorreram depois da finalização do projeto, é a questão de ele está exibindo todo o processo de cálculo da frequência das pesquisas por cidade provavelmente por causa da forma que o laço de repetição foi implementado.

O software ainda pode ser melhorado e muito ele tem muitos problemas de otimização, mas para um piloto está funcionando bem e atende aos requisitos.

References

Menezes, N. N. C. (2019). *Introdução à Programação com Python*. Novatec, terceira edition.