Relatório do componente EXA-854 MI – Algoritmos
Problema 02 – Sistema para o censo demográfico de 2020

Rita Kassiane Santos dos Santos

Engenharia de Computação - Universidade Estadual de Feira de Santana (UEFS)
Caixa Postal 252 e 294 – 44.036-900 – Feira de Santana – BA – Brasil

ritakassiane.t.i@gmail.com

1. Introdução

Desde a Guerra Fria, o crescimento exponencial da tecnologia e a valorização da ciência foram os fatores principais para o desenvolvimento da humanidade. O investimento direcionado a essas áreas abriu caminhos para uma "era" onde o aumento do conhecimento é diretamente proporcional à automatização de processos. Nesse contexto, é pertinente afirmar que a eficiência do trabalho exercido por uma máquina é incomparável com a humana em termos de gasto de tempo e lucro, no entanto, algumas tarefas da máquina ainda precisam da racionalidade singular humana para operar de maneira coerente, e vice-versa.

Diante disso, o IBGE decidiu utilizar a tecnologia para automatizar seus sistemas. Para tanto, solicitou o projeto piloto de uma "calculadora de estatísticas" aos estudantes do MI de Algoritmos, do curso de Engenharia de Computação da UEFS. O software é desenvolvido utilizando a linguagem Python e tem por objetivo calcular 7 estatísticas (as quais foram estabelecidas pelo IBGE) através de importação de arquivos de texto.

Para cumprir os objetivos solicitados, o código-fonte foi baseado em manipulação de strings de maneira manual, ou seja, utilizando conceitos mais elementares da programação, de forma a implementar a menor quantidade de métodos de lista possível.

2. Metodologia

Depois de diversos problemas de interpretação, finalmente foi estabelecido o objetivo final do software a ser desenvolvido: calcular estatísticas utilizando arquivos de texto como dados de entrada. Posteriormente, foi feita uma análise desses arquivos e do questionário, (os quais foram disponibilizados em conjunto à folha do problema) a fim de reconhecer padrões e encontrar as possíveis maneiras de resolução. Com isso, foi possível perceber a semelhança entre a estrutura dos dados do arquivo com uma matriz matemática.

Depois de fazer um estudo mais profundo sobre listas e manipulação de strings em Python, o grupo discutiu a ideia do raciocínio acerca do uso da matriz, de maneira a esclarecer a sua definição (uma lista com outras listas dentro) e estabeleceu como fato a utilização dela, uma vez que foi percebido o seguinte padrão: cada linha seria equivalente a um domicílio assim como cada coluna a uma pergunta do questionário.

A utilidade da matriz esta justamente no ponto onde, através da manipulação das strings dentro das listas, pode-se conseguir obter os dados necessários para o calculo das estatísticas. Para isso, é necessário primeiramente, importar o arquivo de texto (através do método *open)*, lê e transformar cada linha dele em uma lista (através de nomeDaVariávelQueAbreOArquivo.*readlines)*. Posteriormente, utiliza-se um *for* que percorrerá por cada lista criada e através dos métodos S*trip* e *Split,* desconsidera o "/n" que foi gerado ao quebrar o arquivo em linhas, e separa cada string tirando o ";", respectivamente. A cada linha tratada, adiciona-se esta a uma lista vazia, que será a matriz utilizada para as análises e manipulações futuras.

Todavia, antes de começar a utilizar a matriz para o calculo das estatísticas, é necessário levar em conta a existência do técnico e da cidade onde ele esta aplicando a pesquisa através de um módulo de cadastro, (esses são denominados por tecnicosIBGE.txt e regiões.txt, respectivamente), e do fluxo de resposta o qual deve ser validado baseando-se no questionário fornecido.

Para isso, foi criada uma função a qual fora dividida em três pontos:

1. Validação da matrícula do técnico
2. Validação da região onde o técnico esta realizando a coleta
3. Validação do fluxo de respostas da coleta

A validação da matriz ocorre de maneira gradual de forma a validar uma lista por vez, seguindo o seguinte raciocínio:

Primeiramente um *for* irá percorrer cada lista da matriz de maneira a pegar apenas o primeiro elemento dela, o qual é equivalente a matricula do técnico. Posteriormente, através do método *in*, deve-se procurar o técnico no módulo de cadastro tecnicosIBGE.txt atribuindo este procedimento a uma variável. Caso a variável retorne *True* aquela matrícula existe no arquivo txt em que ela foi procurada, logo, o técnico esta validado.

Depois da validação do técnico, o mesmo procedimento é repetido para a região e posteriormente ao fluxo. Caso a lista tenha passado nas três validações, ela é adicionada a uma *matriz* vazia, a qual será utilizada para manipulação.

2.2 Cálculos das estatísticas

O problema solicitou o calculo de sete estatísticas, as quais são respectivamente:

1. Números de domicílios utilizados para a coleta;

O número de domicílios é a quantidade de linhas do arquivo e cada lista dentro da matriz equivale a uma linha deste. Portanto, para calcular a primeira estatística, basta utilizar o método *len* para descobrir o tamanho da lista validada.

2. Número de domicílios particulares que já estão pagos, quantos ainda estão pagando e alugados;

Haverá três variáveis igualadas a zero, as quais servirão de contador para domicílios particulares pagos, pagando e alugados.

Para o calculo dessa estatística, um *for* percorrerá a matriz validada de forma a conferir se na posição equivalente à pergunta 2.01 (a qual fornece a informação desejada sobre o

domicílio) existe a string *1, 2 ou 3*. Em cada caso, adicionará 1 ao contador correspondente a este. Por exemplo, se nessa posição existe *'1'* adicione 1 ao contador de domicílios já pago.

3. Quantos domicílios por cidade possuem banheiro e quantos não possuem;

Inicialmente foi feita uma matriz com três elementos, os quais são respectivamente: uma lista contendo o código IBGE, uma lista do tamanho da anterior onde todos os elementos são zero e uma lista com as mesmas características da anterior. A lista localizada na posição 1 e 2 da matriz funcionaram como contadores para a quantidade de domicílios com banheiro e sem banheiro naquela cidade, respectivamente.

Posteriormente, um *for* percorrerá a matriz validada indo na posição que se trata da quantidade de banheiros do domicílio. Caso esta seja diferente de 0, contabiliza-se +1 na lista 1 da matriz na posição equivalente ao contador inicializado pelo *for* no momento.

Para alterar o código IBGE, foi feita uma função que relaciona este com o nome da cidade através do arquivo regiões. Para tal, um *for* percorre a posição 0 da matriz de três elementos, e altera um código por vez.

4. A forma mais comum de abastecimento de água por cidade;

A forma mais comum de abastecimento de agua por cidade é a *moda* de uma lista. Para isso, foi utilizado o mesmo raciocínio da estatística três, no entanto, na posição 1 havia listas dentro de lista dentro de lista.

5. O percentual de domicílios por cidade que ainda não possuem energia elétrica;

Para tal calculo, utiliza-se o mesmo raciocínio da estatísticas três. No entanto, ambas diferem no retorno dado pela função uma vez que, o valor que será retornado nessa é calculando a porcentagem.

6. O percentual de moradores que participaram da entrevista por cor ou raça.

Para tal calculo, utiliza-se o mesmo raciocínio da estatística dois. No entanto, ambas diferem no retorno dado pela função uma vez que, o valor que será retornado nessa é calculando a porcentagem.

7. A região com maior número de municípios pesquisados.

Haverá cinco variáveis inicialmente igualadas a zero, as quais servirão como contadores para cada região. Um *for* percorrerá a matriz validada indo até a posição que fornece o código IBGE e analisará o primeiro número o qual diz a qual região aquela cidade pertence. Identificando a região, ele somará +1 ao contador relacionado a ela.

Posteriormente à contabilização, cria-se uma lista de cinco elementos, em que cada um deles é dos contadores das regiões. *Exemplo: lista = [norte, sul, sudeste, nordeste, centro-oeste]*. Com isso, um *for* irá percorrer essa lista, comparando os valores de cada posição dela a outro contador (o qual é inicializado com 0) e guardando o maior valor entre os dois na variável do contador.

Por fim, o valor final do contador será o maior da lista, logo a posição dele na lista de contadores, irá fornecer a região com maior número de municípios pesquisados.

3. Resultados e Discussões

Para a utilização eficaz e correta do software desenvolvido, o usuário deve manter os arquivos no mesmo diretório do programa, levando em conta o formato dele, (o qual deve ser *.txt*), e seus os nomes (os quais devem permanecer: *Pesquisa, tecnicosIBGE e regioes*).

Considerando que o software deve ser bem modularizado, as funções foram criadas e organizadas de maneira estratégica uma vez que assim será possível chama-las dentro de outras funções ou reutiliza-las. Nesse contexto, foi criada uma função para cada estatística e criação de matrizes (sejam elas com arquivo inteirou, ou apenas algumas colunas). No entanto, existe mais de um modo de gerar listas dentro de outras listas. Nesse software, foi utilizado o mais rápido: através da soma direta (uso do +=). Todavia, poderia ter realizado o mesmo procedimento através do uso do método append ou identificando o tamanho da lista e adicionando o que deseja na ultima posição.

A fim de melhorar as informações visualizadas pelo usuário, otimizando a interação deste com o programa a fazendo ser mais agradável e intuitiva, foi utilizada a biblioteca *Pandas* para a saída de dados (*printar* as matrizes). Para isso, foi utilizado o método *DataFrame*, o qual organiza listas dentro de lista numa visualização em linhas e colunas, como uma matriz matemática.

4. Conclusão

Com o problema proposto, além da valiosa experiência adquirida no trabalho em equipe, foi possível obter conhecimentos mais elementares da programação, sobretudo no âmbito da manipulação de strings em listas e criação de matrizes em Python. Apesar do lento desenvolvimento do grupo na compreensão do problema, as discussões feitas nas sessões tutoriais foram essenciais para a resolução e o desenvolvimento de um software eficiente.

O Software cumpre o objetivo solicitado pelo problema, no entanto, pode ser implementada maneiras de optimização da interação deste com o usuário e tratamento de erros nos pontos onde esses estão previstos - como o erro que ocorre quando o arquivo não é txt, não esta no mesmo diretório do programa ou não é nomeado do jeito que deveria (*Pesquisa, tecnicosIBGE, regiões)*. Tais problemáticas poderiam ser solucionadas com o aumento do conhecimento na área de programação, o qual fosse suficiente para a criação de uma interface gráfica que facilitaria a relação usuário-software, sobretudo na importação do arquivo.

5. Bibliografia utilizada

Programação Básica - Introdução em Python. Disponível em: <https://www.docdroid.net/SI5rExq/programacao-basica-introducao-em-python.pdf> Acesso em 15 de agosto de 2019.

Python Progressivo – Curso Completo. Disponível em: < https://www.pythonprogressivo.net/> Acesso em 6 de agosto de 2019

Pandas para Análise de Dados. Disponível em: < https://minerandodados.com.br/python-para-analise-de-dados/> Acesso em 26 de julho de 2019