

Problema 03: Manutenção do histórico de problemas do MI de algoritmos do Curso de Engenharia De Computação da UEFS

Uellington Da Conceição Damasceno

¹Curso de Engenharia de Computação – Universidade Estadual de Feira de Santana (UEFS)
Av. Transnordestina, s/n - Novo Horizonte, Feira de Santana - BA, 44036-900

uellington99@gmail.com

1. Introdução

Hoje em dia, há diversos tipos métodos de ensino, que se diferenciam bastantes do método tradicional, onde simplesmente o conteúdo passa-se primeiro pelo professor, para que este, depois venha a transmiti-lo ao aluno. O PBL(do inglês: aprendizado baseado em problemas), é um dos métodos é diferente dos tradicional, onde o conteúdo deve-se ser adquirido pelos alunos, através da resolução de problemas e os professores se tornam orientadores.

A UEFS(Universidade Estadual de Feira de Santana) e um instituição se ensino que adotou esse tipo de metodologia de ensino, no curso de engenharia da computação. A cada semestre é apresentado para os alunos problemas que devem ser resolvidos. Problemas esses que são elaborados e analisados por um grupo de professores. Os problemas apresentados aos alunos a cada semestre, devem ser diferentes, por isso deve haver um meio de arquivar esses problemas, para que a cada semestre os professores possam consultar esses problemas, evitando assim que haja problemas repetidos. Além de se poder manter um histórico do que se foi realizado nesses semestres.

Todos os problemas de semestres passados, acabavam sendo arquivados separadamente em acervos pessoais dos professores que participavam da elaboração do problema e muitos acabavam de perdendo. Assim para facilitar a consulta e a centralização do histórico de problemas apresentado para os alunos, foi solicitado a construção de um software que possa organizar, arquivar de forma centralizada e facilitar a consultado desses problemas por parte dos professores.

Assim foi desenvolvido um programa em Linguagem de programação C, que recebe como entrada um arquivo de texto com os problemas dos semestres passados, organiza-os de forma decrescente em relação ao ano e semestre, possibilita a inserção de novos problemas, a procurar de problemas, e gera um arquivo de saída contendo os problemas organizados.

O programa utiliza em sua estrutura diversas funções, afim de modularizar para melhorar o seu desempenho, melhorar a sua organização, possibilitando assim, facilitar sua manutenção e posteriormente, se necessário, implementa-lo, com o intuito de se operar o programa de outras formas. Esse programa foi desenvolvido somente para a plataforma Windows.

Para facilitar o entendimento esse relatório foi separados em 4 partes. sendo elas: Metodologia, Resultados e Discussões e Conclusão.

2. Metodologia

Durante o desenvolvimento do software, ocorreram algumas reuniões que tinham como principal objetivo, solucionar problemas que apareceram e que poderiam aparecer durante a implementação do software. Dentre todas as ideias comentadas durante essas reuniões. Pode-se destacar: A utilização da árvore binária, método de ordenação da lista, soluções para erros gerados pela codificação (UTF-8) do arquivo de entrada e algumas formas de ler o arquivo de entrada.

2.1. Utilização da árvore binária

Para alcançar todos os requisitos mínimos exigidos pelo software, notou-se que era necessário ordenar a lista e buscar determinado elemento na mesma. Diante disso, surgiu a necessidade de pesquisar alguns algoritmos de busca e ordenação.

A ideia da Utilização da árvore binária surgiu quando um dos participantes da reunião (entusiasta da velocidade e eficiência) se empolgou com a agilidade dessa estrutura, tanto na busca, quanto na inserção de um novo elemento. Baseando-se somente na quantidade de verificações e tempo de execução o percussor da ideia não levou em consideração as situações em que melhor se aplicava tal estrutura de dados.

Apesar de ter existido algumas discussões essa ideia foi abandonada. Pois, a árvore binária é método de busca baseado em duas condições. O que inviabilizava a utilização da mesma em um algoritmo que precisa ser organizado seguindo mais de um critério (O que acontecia com o software em questão).

2.2. Ordenação da lista

Antes de falar sobre a solução para o ordenamento da lista. Deve-se comentar sobre o algoritmo de ordenação quicksort, que foi proposto pelo mesmo indivíduo que propôs a utilização da árvore binária.

O quicksort é um algoritmo de ordenação baseado em comparações, que tem como complexabilidade $O(n^2)$ no pior caso e $O(n \log n)$ no melhor caso. Ele adota a estratégia de divisão e conquista, essa estratégia consiste em reorganizar as chaves de modo que as chaves que possuem o menor valor precedam as chaves com valores maiores. A Figura 01 mostra a aplicação do quicksort.

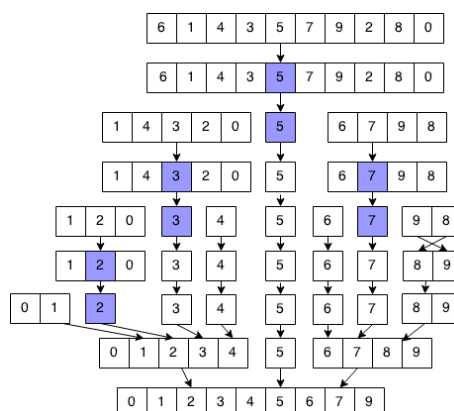


Figura 1. Fonte Próprio autor

Apesar de ser um método de ordenação válido, ficou decidido que a melhor maneira de fazer a ordenação da lista era fazer a inserção de modo organizado. Fazendo com que o novo elemento vá diretamente para a posição que deve ficar. A figura 02 mostra o algoritmo da inserção organizada do elemento.

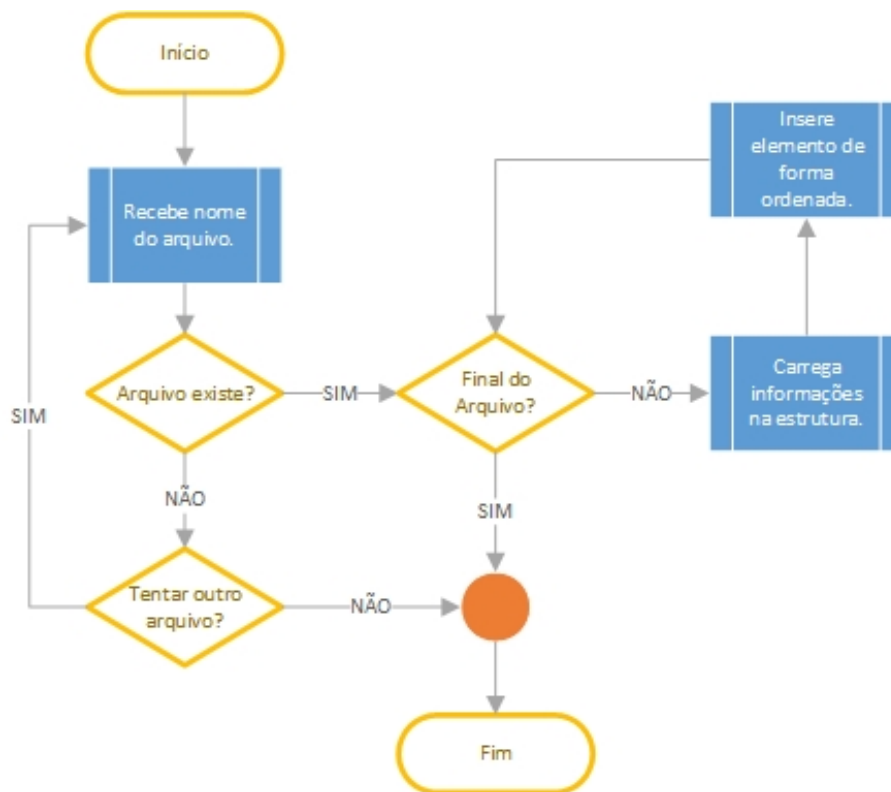


Figura 2. Fonte Próprio autor

2.3. Codificação do arquivo de entrada

Um dos problemas que surgiu com relação da leitura do arquivo foi a codificação a qual o mesmo se encontra. Por algum motivo o arquivo de entrada possui a codificação padrão UTF-8, o que impossibilita a leitura correta do arquivo. Pois, esse padrão possui caracteres a mais.

Como solução para esse problema algumas ideias foram propostas. Sendo elas: Desenvolver um algoritmo de verificação de caractere e pedir para que o usuário mude manualmente o padrão do arquivo de entrada.

A primeira ideia consiste em desenvolver um algoritmo que verifica os 3 primeiros bytes do caractere e se porventura esses bytes corresponderem ao padrão UTF-8 o algoritmo deveria fazer a conversão para ASCII (American Standard Code for Information Interchange) e ao final gerar um novo arquivo.

Por escolha individual essa ideia não foi adotada. Pois, a ideia de fazer com que o usuário tocasse o padrão manualmente aparenta ser mais atrativa para o desenvolvedor do software em questão.

Para o bom funcionamento do software. O usuário deve converter o padrão do arquivo de entrada manualmente. Para fazer isso o usuário deve seguir os seguintes passos.

- Passo 01: Abrir o arquivo utilizado o bloco de notas;
- Passo 02: Ir até o botão “arquivo” no canto superior do bloco de notas;
- Passo 03: Pressionar “salvar como...”;
- Passo 04: Ir em “codificação” (canto inferior direito);
- Passo 05: Mudar para ASCII;
- Passo 06: Salvar mudanças;

É possível que apareça um dialogo de confirmação. Para resolver isso aperte “ok”. Como o padrão alterado o arquivo pode ser lido utilizado normalmente.

2.4. Definição de requisitos

O software em questão tem como principal objetivo receber um arquivo de entrada que possui informações dos PBL anteriores que podem estar ou não desorganizados e com isso organiza-los se necessário e ao final gerar um arquivo de saída todos os problemas organizados de modo decrescente. A figura 03 mostra de modo superficial o funcionamento do software. Mas, para que exista um bom funcionamento do programa o arquivo

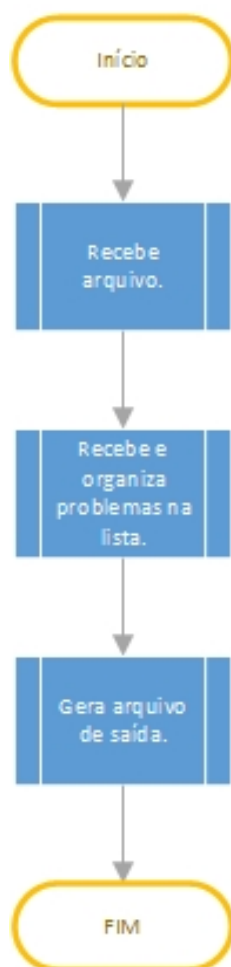


Figura 3. Fonte Próprio autor

de entrada deve seguir o seguinte padrão de layout:

- Primeira informação: Semestre;

- Segunda informação: Número do problema;
- Terceira informação: Ano do problema;
- Quarta informação: Título do problema;
- Quinta informação: Descritivo do problema;

Todos devem est separados por um espaço que corresponde ao tamanho de um “Tab”. Caso exista a inserção de um arquivo que foge desse padrão descrito anteriormente pode acontecer graves falhas na execução do software. Pois, não foi implementado nenhum método de validação de layout.

Vale ressaltar que o arquivo de saída (gerado pelo software) é compatível com esse layout. Portanto é possível utiliza-lo também como arquivo de entrada. Entretanto, deve-se observar que, caso exista alteração no conteúdo o arquivo será sobrescrito.

2.5. Modularização

O software em questão possui 28 funções que podem ser divididas em duas categorias diferentes. Sendo elas: funções acopladas e não acopladas.

2.5.1. Funções Acopladas

Nessa categoria se encaixa todas as funções de design que são responsáveis por desenhar bordas ao redor do texto. Além das funções gotoxy e as funções que são responsáveis por permitir ao usuário interromper, voltar ou prosseguir com um comando.

2.5.2. Funções não acopladas

Fazem parte dessa categoria todas as funções relacionadas a leitura, processamento e exportação de arquivo, bem como as funções responsáveis por mostrar o menu, receber, validar e retornar a escolha do usuário.

2.6. Responsabilidades da função principal

No software em questão a função main tem como principal caracteriza redirecionar o fluxo de execução do algoritmo. Entretanto, existe dois momentos em que a main executa um processamento diferente sendo eles: O momento de preencher as estruturas com as informações do arquivo e o momento em que o usuário deve inserir um novo problema. Isso ocorre porque a metodologia utilizada para executar tais procedimentos foi pensada para evitar o acoplamento entre funções.

2.7. Gotoxy

Dentre todas funções que constituem o software, deve-se destacar a utilização da função gotoxy. Ela recebe como parâmetro dois valores inteiros que corresponde a uma coordenada no terminal e após a invocação da mesma o curso é movido até as coordenadas recebidas

Essa função é a única do software que teve o código retirado da internet. Por tanto, é compreensível que ela não pode ser utilizada com um dos critérios de avaliação.

3. Resultados e Discussões

Para utilizar o software, o usuário deve importar um arquivo de entrada condizente com as características ditas no tópico de validação de entrada desse relatório. Em seguida o software irá exibir o menu principal, possibilitando que ele escolha entre as seguintes opções:

- Visualizar conteúdo;
- Adicionar novo problema;
- Buscar problema;
- Gerar novo arquivo;
- Tentar com novo arquivo;
- Sair;

3.1. Visualizar conteúdo

Essa função foi elaborada para possibilitar ao usuário verificar o conteúdo do arquivo inserido. Entretanto, o conteúdo que será mostrado não está exatamente igual ao encontrado no arquivo. Pois, de acordo com a metodologia utilizado para ordenar a lista os itens já são inseridos na posição correta. E por esse motivo, o conteúdo que o usuário irá visualizar equivale aos problemas do arquivo de entrada prontos para serem exportados.

3.2. Adicionar novo problema

Essa função faz parte dos requisitos mínimos do software. Com relação a inserção de um novo problema. Pode-se afirmar que ela não é feita por uma função. Mas sim, 5 funções. Sendo elas:

- ler ano;
- ler semestre;
- ler problema;
- ler título e descritivo;
- insere ordenado;

É fato que poderia ser utilizado apenas uma função para preencher a estrutura problema com as informações digitadas pelo usuário. Mas, por questão de design e segurança ela foi separada de modo que não ficasse grande.

3.3. Função ler ano

Como o próprio nome diz, essa função é responsável por receber o ano do problema. Mas, além disso ela também limita a inserção de valores discrepantes. Caso exista a inserção de um valor discrepante (menor que 2003 e maior que 2018) a função irá apresentar uma mensagem bem-humorada e permitirá ao usuário inserir novamente um ano aceitável.

3.4. Função ler semestre

Essa função tem uma funcionalidade parecida com a anterior. Mas, possui algumas características que devem ser observadas. Pois, nesse caso o usuário não precisa digitar o semestre completo como por exemplo “2017.1”. Mas sim apenas o ultimo dígito.

Essa metodologia foi implementada por causa de algumas falhas que estavam ocorrendo com relação a validação do semestre digitado. E como solução para o problema de validação se fez necessário a implementação de uma “solução de engenharia”.

Essa solução consiste em pegar o valor digitado pelo usuário e dividir por 10 e somar com o ano digitado na função anterior. E assim consequentemente conseguindo um valor decimal correspondente ao semestre.

3.5. Função insere ordenado

A função insere ordenado tem como principal função inserir de modo ordenado (decrescente) um novo problema que é passado por parâmetro, sendo ele digitado pelo usuário ou importado do arquivo de entrada. Essa função consiste basicamente em uma serie de verificações que estão envoltas por um laço de repetição responsável por percorrer toda a lista em busca da posição correta. A figura 04 mostra o fluxograma do algoritmo de ordenação;

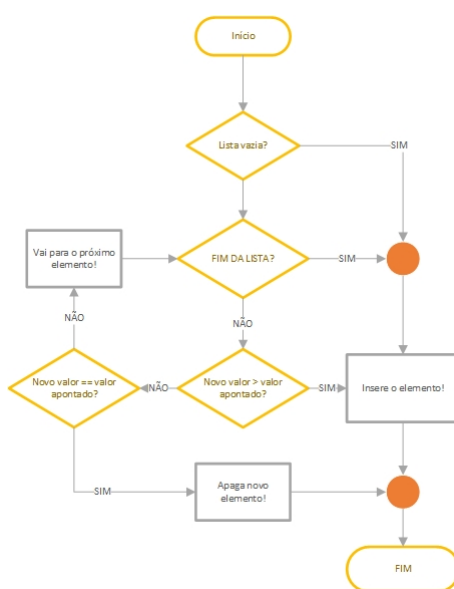


Figura 4. Fonte Próprio autor

Como mecanismo de comparação para a inserção dos itens na lista foi necessário criar uma variável dentro da estrutura chamada de “valor de ordem” e com base no valor que contido nela a célula é inserida em determinado lugar da lista.

Para obter o valor necessário para ordenação e necessário que o semestre e o problema tenham sido preenchidos e ou importados corretamente. Pois, o valor de ordem equivale ao resultado da seguinte expressão: $\text{valor de ordem} = \text{semestre} + (\text{problema} / 100)$;

Exemplo: se por ventura o problema for do semestre 2017.1 e tiver o 2 como número do problema o valor de ordem desse problema vai ser 2017.12 e assim reduzindo consideravelmente a quantidade de verificações necessárias para a inserção ordenada.

A função ler título e descritivo não foi comentada. Porque não contém nada de novo na implementação. Já a função ler problema segue o mesmo padrão que a ler semestre limitando apenas a inserção de valores de problemas menores que 0 e maiores que 10.

3.6. Buscar no problema

Após selecionado a opção “buscar problema” o usuário irá se deparar com um sub menu de busca que possibilita que ele liste problemas seguindo os critérios desejados. Dentre esses critérios estão:

- Listar por ano;
- Problema específico;
- Todos os semestre;

Após selecionar um dos métodos de busca o software irá pedir o valor do semestre e ou o ano. E com base no valor digitado ele irá percorrer toda a lista procurando o elemento digitado. Se por acaso não existir esse elemento ele irá imprimir uma mensagem informando ao usuário e ao final voltará para o menu de busca;

3.7. Gerar novo arquivo

Essa função tem como principal função exportar um arquivo chamado “log org.txt” que possui como conteúdo as informações devidamente ordenadas.

Vale também lembrar que o arquivo exportado contém o layout compatível com o necessário para a importa-lo novamente ao software.

Já função tentar outro arquivo permite ao usuário importar outro arquivo sem que exista a necessidade de fechar o software.

4. Conclusão

Após todo o trabalho de desenvolvimento, o programa chegou em um ponto do qual consegue executar razoavelmente todos os requisitos mínimos necessário. Requisitos esses que variam desde um simples carregamento do arquivo, ordenação, inserção de item na lista até gerar um arquivo de saída com as informações pré-definidas.

Na atual versão do programa não foi encontrado nenhum bug. Entretanto, é possível que exista algum problema ainda não descoberto, que pode ser desencadeado depois e alguma sequência de inserções de entradas ou arquivos inválidos.

Para um melhor desempenho do programa, poderia ser implementado o uso de gerenciamento de arquivos de entrada. permitindo ao usuário inserir novos arquivos e compara-los entre si.

5. Referencias

Nenhuma referencia foi consultada.