



UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB

SISTEMAS DE INFORMAÇÃO – SI
HUMBERTO JOSÉ DA SILVA JÚNIOR

RELATÓRIO DE ANÁLISE DE ALGORITMOS REFERENTE AO TRABALHO DA TERCEIRA
AValiação DA DISCIPLINA DE ESTRUTURAS DE DADOS I

1. Resumo

O presente relatório descreve as resoluções dos exercícios propostos no trabalho complementar da terceira avaliação da disciplina de Estruturas de Dados I. Os algoritmos de solução foram desenvolvidos no paradigma imperativo, utilizando a linguagem de programação C. Os mesmos foram desenvolvidos com os conhecimentos apresentados na disciplina.

2. Introdução

Este relatório tem como objetivo descrever o trabalho complementar à terceira nota da disciplina Estrutura de Dados I, do curso superior de Sistemas de Informação ofertado pela Universidade Federal do Piauí, Campus Helvídio Nunes de Barros, em Picos - PI.

O presente trabalho tem como propósito proporcionar ao discente [do curso] conciliar os conhecimentos teóricos à prática por meio das resoluções de enigmas matemáticos e/ou regras de negócios não triviais, em diferentes níveis de complexidade.

As atividades propostas no supramencionado trabalho, foram solucionadas utilizando o paradigma imperativo, especificamente na linguagem de programação C, stack adotada para fins de estudo da disciplina.

3. Sessões Específicas

a. Informações técnicas

Foi utilizado um notebook de marca Acer, modelo Aspire 3, com um processador Intel Core i5 de sétima geração, com oito gigabytes de memória ram, quinhentos e doze gigabytes SSD (Solid State Drive), um terabyte de armazenamento HDD (Hard Disk Drive) com a distribuição Zorin OS, do sistema operacional linux.

Para o desenvolvimento dos algoritmos, foi utilizado o Visual Studio Code - como editor de códigos, e o GCC na versão 9.3 - como compilador.

b. Funções comuns

Algumas funções são comuns nos códigos, ou seja, são chamadas mais de uma vez em um ou mais scripts.

A função nomeada de "init" trata em receber uma estrutura [lista circular] e iniciá-la, ou seja, atribuir ao início e fim - da mesma, o valor de nulo (NULL).

Nomeada de "insert", a função tem como objetivo inserir elementos de forma ordenada na estrutura. A função recebe dois parâmetros, o primeiro é um ponteiro da lista, e o segundo o dado que será inserido.



UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB

SISTEMAS DE INFORMAÇÃO – SI
HUMBERTO JOSÉ DA SILVA JÚNIOR

Na função, é alocado um endereço na memória, do tipo da lista, e em seguida é verificado se o mesmo é diferente de nulo (NULL), caso a condição seja atendida, é verificada se a variável “aux” - que inicialmente recebe o início da lista - é igual a nulo (NULL), caso seja, isso representa que a mesma está vazia, ou seja, o dado será o início da mesma, caso a condição não seja atendida, é verificado se o mesmo é menor que o elemento do início da lista, caso for, é inserido no início, caso contrário é verificado se o mesmo é maior que o último elemento, caso seja ele será inserido no final da lista, por fim, caso ela não atende nenhuma das verificações anterior, o mesmo será inserido no meio. Um laço é implementado, tendo condição de parada quando o dado que será inserido for maior que o percorrido, dentro do laço, é realizada apenas a passagem para o próximo elemento. Assim que o laço for interrompido, o novo dado será inserido entre o anterior e o próximo.

A função “delete” tem como principal objetivo remover um elemento da lista, a mesma também recebe dois parâmetros, o ponteiro para a lista e o dado a ser removido.

Inicialmente a função verifica se o elemento a ser removido é o primeiro da lista, caso seja, o mesmo é removido, caso contrário é verificado se é o último elemento da lista, caso contrário o mesmo a ser removido será verificado se estar no meio da lista, um laço fica percorrendo enquanto o elemento procurado seja diferente ao do aux e aux seja diferente do início da lista, após é verificado se o mesmo é igual, caso não seja o mesmo não pertence a lista.

A função nomeada de “showAll” foi implementada com a finalidade de mostrar todos os elementos da lista. A mesma cria uma variável ponteiro, do tipo “tlist” que inicialmente recebeu o início da lista, e em seguida é verificado se o mesmo é diferente de nulo (NULL), caso seja atendida, um laço de repetição “do while” que tem como condição de parada, quando a variável “aux” for diferente do início da lista “l.s”. No laço de repetição, é chamado a função “showOne” e passando o dado do “aux” como parâmetro para a função, em seguida, “aux” passa a receber o próximo elemento do mesmo.

A função “showOne” trata em imprimir um elemento, a mesma recebe um parâmetro do tipo “tdado” e em seguida realiza imprime todos os dados do dado passado.

c. Exercício 01

Na primeira questão é solicitado que seja desenvolvido um programa que cadastre polinômios em três variáveis, que pode ser representado em uma lista circular, onde cada nó representa um termo, e a posteriormente seja realizado a soma de dois polinômios, multiplicar dois polinômios, avaliar, imprimir um polinômio, imprimir todos e



UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB

SISTEMAS DE INFORMAÇÃO – SI
HUMBERTO JOSÉ DA SILVA JÚNIOR

calcular a derivada parcial de um dos polinômios, em relação a qualquer uma das variáveis.

Para este problema, foram desenvolvidas as funções “somar_polinomios”, “imprimirLista”, “avaliar_polinomio”, “imprimir_polinomios” e “derivada_parcial”.

A função para somar dois polinômios, “somar_polinomios”, recebe como parâmetro o array de polinômios e os dois índices em que os mesmos se encontram. Inicialmente cria duas variáveis do tipo nó, no qual recebe o início de cada uma das listas. Em uma laço de repetição, que executa três vezes, é verificado se existe a variável nos dois polinômios, caso exista, é verificado se a potência do polinômio primeiro é igual a do segundo, caso seja atendida, é inserido em uma lista auxiliar, somando as duas potências e somando os valores das constantes, caso não exista a variável em nenhum dos polinômios, é verificado se existe a variável apenas no primeiro polinômio, o mesmo é inserido, e depois verificado no segundo e inserido, caso não exista em nenhum, não é inserido. Após o fim dos condicionais, é incrementado a variável de controle “i” em mais um (i+1) e em seguida verificado os demais termos.

Para avaliar um polinômio, a função “avaliar_polinomio”, inicialmente é lido um valor para cada uma das variáveis e em seguida é calculado o valor dos termos e por último calculado com a constante.

Para calcular a derivada parcial, a função “derivada_parcial” foi desenvolvida, a mesma recebe a lista que foi escolhida pelo usuário, a variável a ser derivada, e o valor da mesma. Em um laço de repetição que percorre todos os termos da lista, é verificado se a variável informada é igual a do termo, caso seja, por meio da função pow, da biblioteca math.h é elevado o valor da potência do termo e por último multiplicado pelo coeficiente.

d. Exercício 02

No exercício número dois (02), é solicitado que seja desenvolvido uma roleta de prêmios, em que tenha uma opção para cadastrar os prêmios e outra opção para sortear um prêmio.

Para este problema, no main foi declarada e iniciada uma variável “flag” que inicialmente recebe o valor de zero “0” para controle da direção do que a lista irá percorrer, ou seja, zero (0) no sentido horário e um (1) no anti-horário.

A função nomeada de “raffle”, foi implementada, a mesma recebe um ponteiro para a lista e um ponteiro para a flag no qual controla o sentido que a lista vai ser percorrida.

A função trata de sortear um número randômico entre um e cem (1 ~ 100) e a partir do mesmo, será a quantidade de vezes que a lista vai “rodar”, ou seja, ficar percorrendo. Assim que o laço encerrar, será decrementada em um (-1) a quantidade do prêmio que parou e em seguida exibido o prêmio sorteado.

Na função é declarado dois ponteiros para o tipo “tlist”, sendo elas: “sAux”, para o início da lista e “eAux” para ao fim. Também são declaradas duas variáveis do tipo inteiro, “numRandom” e “count”, para guardar o número sorteado e o contador de vezes que percorreu a lista, respectivamente.



UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB

SISTEMAS DE INFORMAÇÃO – SI
HUMBERTO JOSÉ DA SILVA JÚNIOR

Inicialmente é verificado o valor da flag - "f", para atribuir aos ponteiros os valores do início e fim da lista. Caso o valor da "flag" seja um (1), a variável que recebe o ponteiro do início vai receber o fim da lista, e a que vai armazenar o fim vai receber o início, isso quer dizer que a lista vai ser invertida.

Posteriormente é verificado se o ponteiro auxiliar usado para armazenar o início é diferente de nulo (NULL), caso a condição seja atendida, um laço de repetição é executado, o mesmo será executado enquanto o contador for menor que o número randômico sorteador ($\text{count} < \text{numRandom}$).

Dentro do laço de repetição, a variável "sAux" recebe o resultado de um condicional do tipo ternário, ou seja, se a flag - "f", for igual a um (1), vai ser atribuído o anterior, caso contrário, o início. Em seguida, é incrementado o contador em mais um ($\text{count} + 1$), e atualizado o início e fim da lista.

Após o laço finalizar, é exibido o prêmio que o cliente sortearou. Caso a flag seja um (1), a variável aux vai receber o próximo elemento da variável "sAux", caso contrário, vai receber o anterior, em seguida é impresso o prêmio é decrementado a quantidade e menos um (-1) e por fim é verificado a quantidade de elementos do prêmio sorteado, se for igual a zero o mesmo é removido da roleta.

4. Resultados das Execução dos Programas

Para comprovar o funcionamento dos algoritmos solucionados em cada questão, foram realizados inúmeros testes a fim de tratar eventuais erros. Uma tabela de duas colunas - Entrada e Saída, demonstra logo abaixo.

A primeira questão solicita que seja realizado um programa que manipule polinômios em três variáveis e seja realizado as seguintes operações: 1 - Cadastrar; 2 - Somar; 3 - Multiplicar; 4 - Avaliar; 5 - Imprimir; um polinômio; 6 - Imprimir todos os polinômios.

ENTRADA	SAÍDA
1 - $2x^2 + 4y^3 + 7z^4 + 3$; $6x^2 + 8y^3 + 1z^4 + 3$; $2x^5 + 3y^6 + 2z^7 + 3$; $2x^3 + 7z^8 + 3$; e $0x^5 + 8y^6 + 5z^7 + 3$	
2 - 0 e 1	$8x^2 + 12y^3 + 8z^4 + 6$
2 - 0 e 2	$2x^2 + 2y^5 + 4z^3 + 3^{67}42^{76}$
5 - 2	$6x^2 + 8y^3 + 1z^4 + 3$
4 - 4 (x=2, y=2 e z=2)	$6x^2 + 8y^3 + 1z^4 + 3$ Resultado: 1155
7- 7 (x=5)	$2x^3 + (2734375) + 3$



UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB

SISTEMAS DE INFORMAÇÃO – SI
HUMBERTO JOSÉ DA SILVA JÚNIOR

A segunda questão, solicita que sejam cadastrados prêmios em uma roleta de sorteio e que sejam sorteados prêmios. 1 - inserir, 2 - Todos e 3 - sortear.

ENTRADA	SAÍDA
1 - Id = 1, Nome = Prêmio 1, Quantidade = 1; Id = 2, Nome = Prêmio 2, Quantidade = 2; Id = 3, Nome = Prêmio 3, Quantidade = 3; Id = 4, Nome = Prêmio 4, Quantidade = 4; Id = 5, Nome = Prêmio 5, Quantidade = 5;	# SUCCESS! # SUCCESS! # SUCCESS! # SUCCESS! # SUCCESS!
2	Id = 1, Nome = Prêmio 1, Quantidade = 1; Id = 2, Nome = Prêmio 2, Quantidade = 2; Id = 3, Nome = Prêmio 3, Quantidade = 3; Id = 4, Nome = Prêmio 4, Quantidade = 4; Id = 5, Nome = Prêmio 5, Quantidade = 5;
3	N RAND = 10 Premio 5
3	N RAND = 4 Premio 2
3	N RAND = 33 Premio 4
3	N RAND = 82 Premio 3
3	N RAND = 51 Premio 3
3	N RAND = 31 Premio 3 #PRÊMIO REMOVIDO

5. Conclusão

É perceptível a importância do trabalho, como já supracitado, em aliar a teoria das aulas com a prática das resoluções dos problemas propostos, que apesar de serem um pouco complexos, foi bem divertido de resolvê-los.

Por fim, após a resolução dos problemas, fica uma sensação de dever cumprido e é notório que o trabalho já supracitado, exige tempo e acima de tudo muita dedicação.



UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB

SISTEMAS DE INFORMAÇÃO – SI
HUMBERTO JOSÉ DA SILVA JÚNIOR