Universidade de Brasília
Departamento de Ciência da Computação

Árvores binárias

Aluno: Hiago dos Santos Rabelo - 16/0124492
Disciplina: Estrutura de dados
Professor: Eduardo A. P. Alchieri

Novembro de 2017

Sumário

1 Introdução

O problema a ser resolvido consiste em se ler um arquivo .txt e armazenar suas informações tanto em uma árvore binária de busca como em uma lista, posteriormente usar essa informação para decodificar uma mensagem codificada em morse.
Por fim, mostrar a diferença entre o tempo de decodificação usando-se listas e usando-se árvores binárias de busca.

2 Implementação

2.1 Estrutura de dados utilizada

As estruturas de dados utilizadas para resolver o problema proposto no trabalho foram: árvore binária de busca e listas.

(i) Árvore binária de busca:

- Tem por função armazenar os caracteres na ordem de seus respectivos códigos em morse, sendo que isso é feito de forma que quando o caractere é inserido na árvore é lido caractere por caractere de seu código morse e para cada simbolo lido é tomado um caminho diferente.
  Tomando-se o exemplo do caractere `A` codificado como: `.-`.
  Logo, temos que acessar o filho da esquerda, depois o da direita e depois o da esquerda, sendo que como chegou ao fim do código especificado devemos colocar o caractere `A` nesse nó. Seguindo-se esse mesmo raciocínio para o restante dos caracteres.

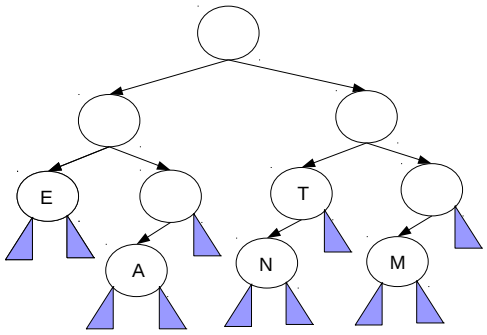

Figura 1: Árvore com chave do código morse

(ii) Lista :

- Tem por função armazenar os caracteres, seus respectivos códigos em morse e um ponteiro para o próximo elemento, sendo que ponteiro do último elemento aponta para NULL.
  Sendo assim, quando for feita a busca para tentar descriptografar a mensagem deverá ser percorrida a lista desde o início até que se seja encontrado o código o qual se está buscando e assim extrair o caractere que corresponde a esse caractere e que se encontra armazenado na mesma posição da lista em que se estava o caractere correspondente.

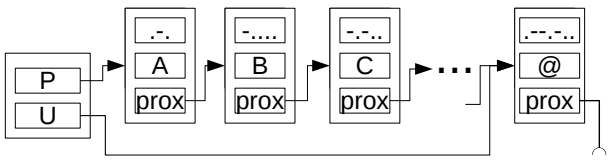

Figura 2: Lista float.

2.2 Principais funções

Temos a seguir os nomes das principais funções, iguais aos colocados no programa, e suas respectivas descrições:

(i) cria_no :

- Aloca espaço de memória para a estrutura do tipo *no* então retornando o endereço do espaço alocado e caso não seja possível retorna NULL, sendo que irá colocar NULL na variável prox dentro da estrutura alocada.

(ii) passa_arquivo_chave_para_arvore:

- recebe como parâmetro o nome do arquivo com a codificação morse e lendo desse arquivo, caso exista e possa ser lido, as codificações e seus respectivos caracteres. Sendo que essa informação é passada para a árvore como descrito acima.

(iii) zera_lista:

- recebe como parâmetro uma lista e atribui ao ponteiro do primeiro elemento e do último o valor NULL.'

(iv) decodifica_usando_arvore:

- recebe como parâmetro o nome do arquivo com a mensagem para ser decodificada e usando a árvore binária de busca para decodificar. Sendo que, lê simbolo por símbolo e percore a árvore para achar o caractere correspondente usando a relação de caso o simbolo seja '.' deverá ir para o filho da esquerda e caso seja '-' então deverá ir para o filho da direita.

(v) decodifica_usando_lista:

- Recebe como parâmetros o endereço da lista de elementos (cada elemento contendo um código e seu respectivo caractere como mostrado na Figura 2) e o nome do arquivo a ser decodificada e percorre uma busca na lista para achar o caractere desejado.

3 Conclusão

Conclui-se que este trabalho requisitou vários conhecimentos adiquiridos na matéria de Algoritmos e Programação de Computadores, tendo como principais dificuldades o armazenamento e leitura da lista.

Por fim, foi possível notar que o programa, que utilizava o algoritmo de árvore binária de busca, para resolver o problema foi consideravelemente mais rápido do que o programa que utilizava lista, uma vez que a seguinte operação

```
1          tempo_medido = clock() - tempo_medido;
```

resultou em 0.000031 segundos sendo utilizada a árvore e 0.000066 segundos sendo utilizada a lista.

Referências

[1] Wikipédia, "Bibtex." https://pt.wikipedia.org/wiki/BibTeX, Janeiro 2009. Acessado em: 16 Novembro 2017.

[2] C. Fernandes, "Guia latex - parte iii: Estruturando e marcando o texto." http://www.vidageek.net/2007/10/29/guia-latex-parte-iii-estruturando-e-marcando-o-texto/, Setembro 2007. Acessado em: 16 Novembro 2017.

[3] G. N. Ramos, "Estruturas de dados não lineares." gnramos@unb.br.

[4] G. N. Ramos, "Algoritmos & ordenação." gnramos@unb.br.

[5] C. Vellage, "A simple guide to latex - step by step." https://www.latex-tutorial.com/tutorials/, Outubro 2017. Acessado em: 13 Novembro 2017.