



Universidade Federal do Amazonas

Faculdade de Tecnologia

Trabalho 3: Árvore PATRICIA

Manaus

2020

Laura Aguiar Martinho - 21952064

Árvore Patrícia

Trabalho prático apresentado à disciplina Algoritmos e Estruturas de Dados como parte dos requisitos necessários para obtenção de nota parcial.

Professor: Edson Nascimento Silva Júnior

Manaus

2020

INTRODUÇÃO

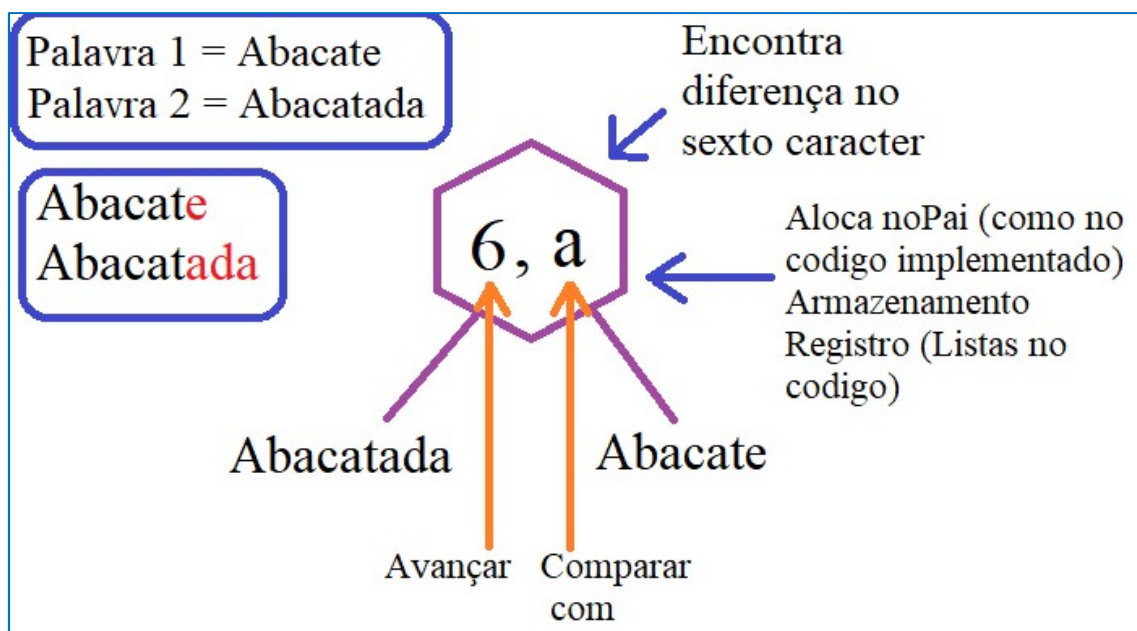
Nas Árvores Trie, definidas em 1960 por Edward Fredkin, termo advindo do inglês, retrieval, cada nó contém informações sobre um ou mais símbolos do alfabeto utilizado. Elas são estruturas indicadas para suportar tarefas de tratamento léxico, como: buscas em dicionários, pesquisas em textos grandes, índices de documentos, entre outros padrões de pesquisa.

A utilização de uma Trie apenas é eficiente se o acesso aos componentes individuais das chaves for bastante rápido. Quanto maior a estrutura mais eficiente uso do espaço. Para driblar o desperdício de espaço com estruturas pequenas foram criadas as árvores PATRICIA

Definida em 1968 por Donald R. Morrison, a Trie Compactada Binária, ou PATRICIA (acrônimo de Pratical Algorithm To Retrieve Information Coded in Alphanumeric), é particularmente útil para tratar de chaves de tamanho variável extremamente longas, tais como títulos e frases.

Também conhecida como RADIX, É uma representação compacta de uma Trie, onde os nós que teriam apenas um filho são agrupados nos seus antecessores. É comum que muitos nós das tries tenham apenas um nó descendente. Isto faz com que as tries tenham um custo elevado de espaço Caminhos que possuem nós com apenas 1 filho são agrupados em uma única aresta. Diferente das Tries não armazena informações nos nodos internos, apenas contadores e ponteiros para cada subárvore descendente.

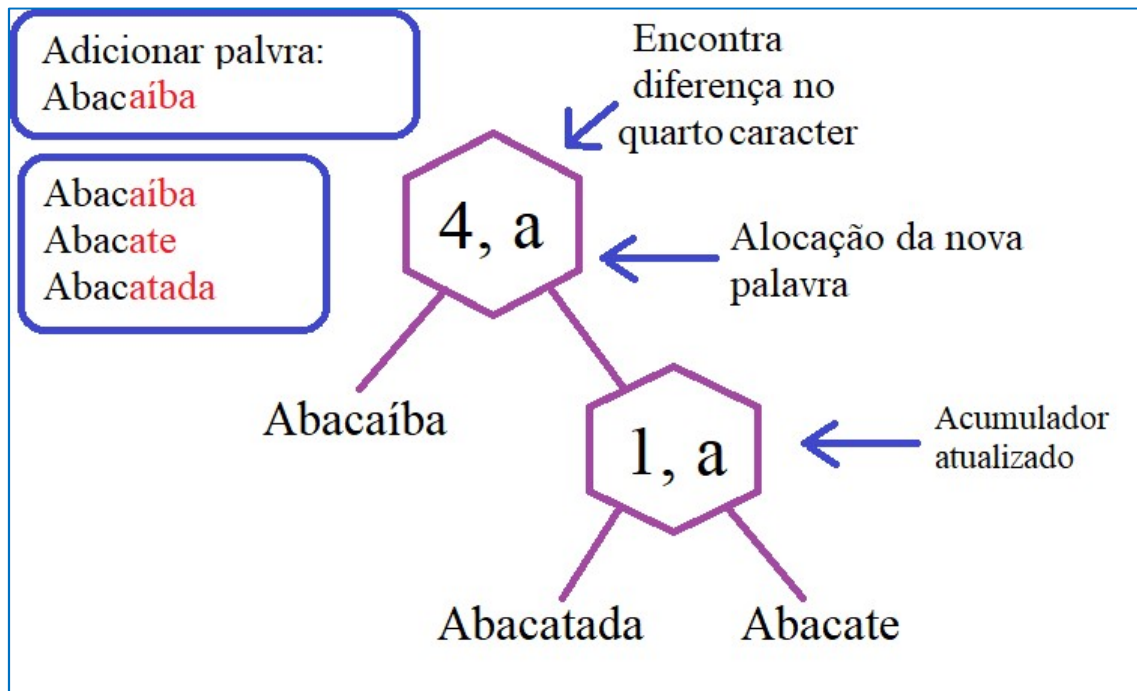
A seguir, esquematização para demonstrar brevemente a dinâmica da PATRICIA:



Exemplo pela alocação das palavras “abacate” e “abacatada”

O campo “avançar” é um registro acumulativo que integra todos os nós, exceto os folhas. Identifica qual a posição do caracter da chave informada que deve ser analisado. O campo “comparar com” apresenta o caracter que deve ser comparado ao caracter da

chave informada como nas árvores binárias de busca, após a análise, se a chave é menor ou igual ao nodo ela é alocada na esquerda ou na direita.



Exemplo pela inserção de nova palavra

Primeiro nó informando pra comparar 4º caracter da palavra. Logo depois, como “t” é maior que “i” desloca-se pra sub-árvore da direita. Então comparam-se agora as ultimas duas palavras.

Mais detalhes no tópico da Implementação do código.

IMPLEMENTAÇÃO

1. Enunciado

O programa foi criado para ler um texto qualquer (arquivo no formato .txt) e imprimir, em ordem alfabética, todas palavras com 3 ou mais caracteres e a(s)linha(s) na(s) qual(is) elas aparecem no texto, desprezando espaços em branco e sinais de pontuação, considerando-os separadores de palavras. Além disso, a leitura deverá converter todas as letras maiúsculas em minúsculas, considerando cada que palavra contém no máximo 20 letras e que não haverá caracteres acentuados.

Exemplo de entrada:

```
Um
Dois
Teste
Testando
Quatro
Ultimo teste
Adeus
```

Saída:

```
Adeus, linha 7
Dois, linha 2
Quatro, linha 5
Testando, linha 4
Teste, linha 3, linha 6
Ultimo, linha 6
```

2. Programa

a. Estrutura

Foram declaradas as estruturas e os protótipos das funções em uma biblioteca.h (arvP.h), funções implementadas no arquivo .c (arvP.c) e a main com leitura dos arquivos e manipulação de caracteres em arquivo .c (mainP.c).

Vale ressaltar que no código, foram manipuladas inúmeras estruturas adquiridas como conhecimento ao longo do curso, nas matérias de Algoritmos e Estruturas de Dados (tanto I quanto II), tais como árvore, vetores, listas, ponteiros, condicionais, entre outros, além de serem incluídas diversas bibliotecas, permitindo que o trabalho fosse um ótimo meio para revisão, consolidação e finalização dos aprendizados desta etapa.

```
/*
****
Biblioteca PATRICIA - Estruturas e prototipos de funcoes
****
*/

//declaração de um tipo String
typedef char* String;

//passando como parâmetro a enumeração dos tipos de NO para ser retornada em funcao futuramente
typedef enum{
    Pai, Folha //noPai (interno) possui filhos, noFolha (externo) sem filhos
}tipoNo;

//estruturas para Lista, usada na implementacao
typedef struct elemento{
    int dado;
    struct elemento *proximo;
}tipoEle;

typedef struct lista{
    tipoEle *primeiro;
    tipoEle *ultimo;
}tipoLista;

//estruturas da arvore e seus nos
typedef struct tNoPAT *tipoPAT;
typedef struct tNoPAT{
    tipoNo tipo;
    //declarar dentro de uma uniao: deixando todos os campos na mesma posição de memória
    union{
        struct{
            short Indice;
            char Caractere;
            tipoPAT Esquerdo, Direito;
        }noPai;
        String Dado;
    }NO;
    tipoLista *linhas; //Guarda as linhas que aparece Chave
}tNoPAT;

//prototipos das funcoes
char Digito(short, String);
int Comparou(char, char);
int EFolha(tipoPAT);
void Ordena(tipoPAT);
void ignoraSimbolo(String, int);
void inserirLista(tipoLista *, int);
void imprimirLista(tipoEle *);
void mostrarPalavra(tipoPAT);
tipoPAT criaPai(short, tipoPAT *, tipoPAT *, char, int);
tipoPAT criaFolha(String, int);
tipoPAT Busca(String, tipoPAT);
tipoPAT InsereAux(String, tipoPAT, short, int);
tipoPAT Insere(String, tipoPAT, int);
tipoLista *criarLista();
```

b. Inserção

Foi usada uma função auxiliar de alocação para cada nodo e função principal para atribuir valores, e a função comparação essencial para o código.

```
//funcao auxiliar insercao, inserir entre chaves
tipoPAT InsereAux(String palavra, tipoPAT arv, short i, int linha){
    tipoPAT no;
    char caractere;
    if (eFolha(arv) || i < arv->NO.noPai.Indice){
        if(eFolha(arv)){
            //se identificou folha, retorna o i-esimo digito do no
            caractere = Digito(i, arv->NO.Dado);
        }else{
            //se nao for folha se torna no com filhos
            caractere = arv->NO.noPai.Caractere;
        }
        //se indice for menor que o pai, cria folha nova
        no = criaFolha(palavra, linha);
        //quando confirma menor ou igual
        if (Comparou(Digito(i, palavra), caractere)){
            //cria espaco pra filhos no noh
            return (criaPai(i, &no, &arv, Digito(i, palavra), linha));
        }else{
            //se nao cria novo pai na arvore
            return (criaPai(i, &arv, &no, caractere, linha));
        }
    }else{
        //verificar comparacoes no pai e palavras
        if(Comparou(Digito((arv->NO.noPai.Indice, palavra), (arv->NO.noPai.Caractere))){
            (arv->NO.noPai.Esquerdo = InsereAux(palavra, (arv->NO.noPai.Esquerdo, i, linha));
        }else{
            (arv->NO.noPai.Direito = InsereAux(palavra, (arv->NO.noPai.Direito, i, linha));
        }
        return (arv);
    }
}
```

```
//funcao inserir valores
tipoPAT Insere(String palavra, tipoPAT arv, int linha){
    tipoPAT no;
    int i;
    if(arv == NULL){
        return (criaFolha(palavra, linha));
    }else{
        no = arv; //faz o no ser raiz da arv
        while (!eFolha(no)){
            if(Comparou(Digito(no->NO.noPai.Indice, palavra), no->NO.noPai.Caractere)){
                no = no->NO.noPai.Esquerdo;
            }else{
                no = no->NO.noPai.Direito;
            }
        }
        i = 0;
        //quando achar o primeiro caracter diferente
        if(strlen(palavra) > strlen(no->NO.Dado)){
            while ((i <= strlen(palavra)) && (Digito(i, palavra) == Digito(i, no->NO.Dado)))
                i++;
            if (i >= strlen(palavra)){
                inserirLista(no->linhas, linha); //inserir valorLinha na linha e retornar raiz
                return (arv);
            }else{ //inserir palavra
                return (InsereAux(palavra, arv, i, linha));
            }
        }else{
            while((i <= strlen(no->NO.Dado)) && (Digito(i, palavra) == Digito(i, no->NO.Dado)))
                i++;
            if (i >= strlen(no->NO.Dado)){
                inserirLista(no->linhas, linha);
                return (arv);
            }else{
                return (InsereAux(palavra, arv, i, linha));
            }
        }
    }
}
```


c. Busca

Para a busca necessária para identificar as chaves e armazenar suas linhas, foi utilizada uma lista para registro.

```
//busca palavra e verifica se existe na arvore
tipoPAT Busca(String palavra, tipoPAT arv){
    if(eFolha(arv)){
        if(strncmp(palavra, arv->NO.Dado, strlen(palavra)) == 0){
            return arv;
        }else{
            return NULL;
        }
    }
    if (Comparou(Digito(arv->NO.noPai.Indice, palavra),arv->NO.noPai.Caractere)){
        return Busca(palavra, arv->NO.noPai.Esquerdo);
    }else{
        return Busca(palavra, arv->NO.noPai.Direito);
    }
}

//funcoes para lista usada no codigo para armazenar linhas contadas
//cria uma lista
tipoLista *criarLista(){
    tipoLista *aux = malloc(sizeof(tipoLista));
    aux->primeiro = NULL;
    aux->ultimo = NULL;
    return aux;
}

//insere valores na lista
void inserirLista(tipoLista *lista, int dado){
    if(lista->primeiro == NULL){
        tipoEle *aux = malloc(sizeof(tipoEle));
        aux->proximo = NULL;
        aux->dado = dado;
        lista->primeiro = aux;
    }else{
        tipoEle *temp = lista->primeiro;
        while(temp->proximo != NULL){
            temp = temp->proximo;
        }
        tipoEle *aux = malloc(sizeof(tipoEle));
        aux->dado = dado;
        aux->proximo = NULL;
        temp->proximo = aux;
        lista->ultimo = aux;
    }
}

//mostra linhas
void imprimirLista(tipoEle *linha){
    if(linha == NULL){
        return;
    }else{
        printf(" - na linha %d;", linha->dado);
        imprimirLista(linha->proximo);
    }
}
```


d. Função principal

Contém as linhas de comando para fazer a chamada e a leitura de um arquivo texto com manipulação adequada das strings.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include "Patricia.h"

int main(int argc, char** argv) {

    FILE *file;
    if((file = fopen(argv[1], "r")) == NULL){
        printf("\nSem arquivo");
    }

    tipoPAT raiz = NULL; int linha = 1;

    while(!feof(file)){

        String palavra = malloc(sizeof(char)*21);
        String frase = malloc(sizeof(char)*100);
        char caractere[21];
        frase = fgets(frase, 100, file);

        //definir um separador que quando surgir na string ela é dividida
        String letras = strtok(frase, " ");

        while(letras != NULL){
            //funcao da biblioteca string strcpy copia caractere no lugar da letras
            strcpy(caractere, letras);
            //percorre o vetor char
            for(int i=0; i<=strlen(caractere); i++){
                //is upper funcao da ctype, checar se e maiuscula (uppercase)
                if(isupper(caractere[i])) caractere[i] = tolower(caractere[i]); //islower para transformar em minuscula (lowercase)
                if(!isalpha(caractere[i])) ignoraSimbolo(caractere, i); //isalpha checa se e letras do alfabeto, se nao, ignorar caracteres
            }strcpy(letras, caractere);

            //apenas palavras com mais de 3 letrass
            if(strlen(letras)>=3) raiz = Insere(letras, raiz, linha);
            letras = strtok(NULL, " ");
        }
        linha++; //contador de linhas
    }
    fclose(file);
    Ordena(raiz);
    return 0;
}
```

CONCLUSÃO

Como já dito, em geral, há grande importância neste trabalho em relação ao entendimento de toda uma etapa referente aos estudos das disciplinas Algoritmos e estruturas de Dados.

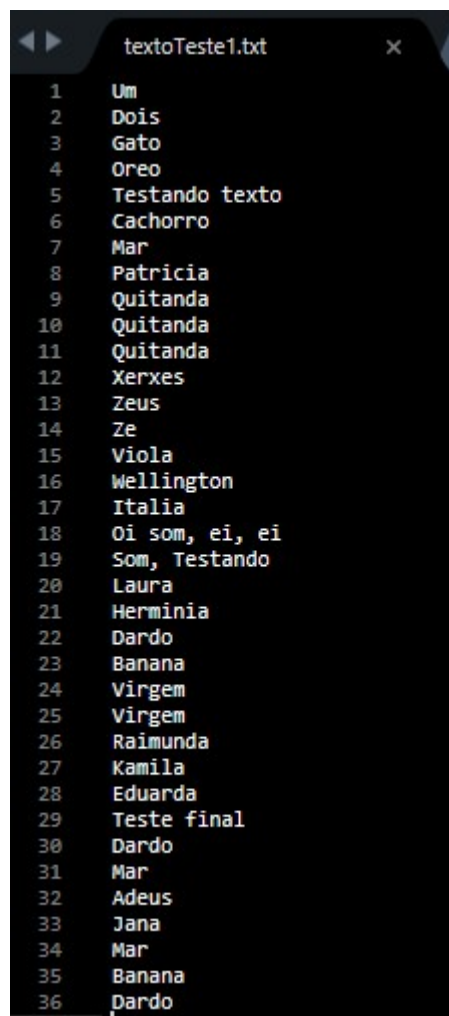
A princípio, entender a forma de execução da PATRICIA foi um desafio. A procura por visualizações ou um norte para implementação foi difícil, mas com a ajuda do professor, numa aula emergencial de dúvidas, muito foi esclarecido, e as dicas ajudaram bastante.

Existiram dificuldades nas questões da manipulação dos arquivos e das strings, como ordenação de caracteres e remoção de caracteres indesejados.

Também na parte da contagem das linhas, foi difícil pensar e concluir uma ideia decisiva para a contagem dentro da estrutura da árvore PATRICIA. Desafios à parte, a implementação foi concluída e o programa executado como desejado.

Chamada: `> prog texto.txt`. Segue execução do programa:

Primeiramente, fiz um texto teste para demonstração do modelo da execução;



```
1 Um
2 Dois
3 Gato
4 Oreo
5 Testando texto
6 Cachorro
7 Mar
8 Patricia
9 Quitanda
10 Quitanda
11 Quitanda
12 Xerxes
13 Zeus
14 Ze
15 Viola
16 Wellington
17 Italia
18 Oi som, ei, ei
19 Som, Testando
20 Laura
21 Herminia
22 Dardo
23 Banana
24 Virgem
25 Virgem
26 Raimunda
27 Kamila
28 Eduarda
29 Teste final
30 Dardo
31 Mar
32 Adeus
33 Jana
34 Mar
35 Banana
36 Dardo
```

```
Microsoft Windows [versão 10.0.19041.572]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Laura>cd desktop\arvore2

C:\Users\Laura\Desktop\arvore2>gcc mainP.c arvP.c arvP.h -o programaPATRICIA

C:\Users\Laura\Desktop\arvore2>programaPATRICIA textoTeste1.txt

adeus - na linha 32;
banana - na linha 23; - na linha 35;
cachorro - na linha 6;
dardo - na linha 22; - na linha 30; - na linha 36;
dois - na linha 2;
eduarda - na linha 28;
final - na linha 29;
gato - na linha 3;
herminia - na linha 21;
italia - na linha 17;
jana - na linha 33;
kamila - na linha 27;
laura - na linha 20;
mar - na linha 7; - na linha 31; - na linha 34;
oreo - na linha 4;
patricia - na linha 8;
quitanda - na linha 9; - na linha 10; - na linha 11;
raimunda - na linha 26;
som - na linha 18; - na linha 19;
testando - na linha 5;
texto - na linha 5;
testando - na linha 19;
teste - na linha 29;
viola - na linha 15;
virgem - na linha 24; - na linha 25;
wellington - na linha 16;
xerxes - na linha 12;
zeus - na linha 13;
C:\Users\Laura\Desktop\arvore2>
```

Agora, execução com os textos indicados no trabalho:

a. Faroeste.txt

```
C:\Users\Laura\Desktop\arvore2>gcc mainP.c arvP.c arvP.h -o programaPATRICIA
```

```
C:\Users\Laura\Desktop\arvore2>programaPATRICIA faroeste.txt
```

```
acabar - na linha 114;  
acabou - na linha 66;  
acabo - na linha 136;  
aceitou - na linha 39;  
acertou - na linha 146;  
achar - na linha 29;  
acontece - na linha 111;  
acreditou - na linha 170;  
agora - na linha 75;  
ainda - na linha 12;  
ali - na linha 18;  
alimentar - na linha 56;  
altar - na linha 16;  
ajudar - na linha 58; - na linha 174;  
ali - na linha 150; - na linha 152;  
alta - na linha 88; - na linha 169;  
amanha - na linha 133;  
amar - na linha 85;  
amigos - na linha 67;  
amor - na linha 138;  
ano - na linha 44;  
antes - na linha 97;  
aos - na linha 24; - na linha 25;  
apareceu - na linha 112;  
apertar - na linha 124;  
aprendeu - na linha 14;  
aprendiz - na linha 45;  
aplaudir - na linha 148;  
aquela - na linha 138;  
aquilo - na linha 18;  
aqui - na linha 38; - na linha 154;  
armar - na linha 108;  
arma - na linha 117;  
armas - na linha 135;
```

armar - na linha 108;
arma - na linha 117;
armas - na linha 135;
arma - na linha 158;
arrependeu - na linha 80; - na linha 165;
asa - na linha 67;
ascendente - na linha 96;
assistir - na linha 144;
ate - na linha 14; - na linha 50; - na linha 55; - na linha 152;
atiro - na linha 160;
atirar - na linha 116;
atirava - na linha 145;
atras - na linha 94;
aumentou - na linha 26;
caboclo - na linha 1;
compositores - na linha 4;
crianca - na linha 11;
cristo - na linha 7;
bagulho - na linha 64;
banca - na linha 91;
bandeirinhas - na linha 148;
bandido - na linha 11; - na linha 75; - na linha 164;
bastardo - na linha 50;
bebedeira - na linha 105;
bestificado - na linha 41;
bisavo - na linha 50;
boiadeiro - na linha 32; - na linha 33;
bolivia - na linha 51; - na linha 109;
bon - na linha 64;
bomba - na linha 91;
bosal - na linha 138;
boto - na linha 91;
boyzinho - na linha 69;
brasilha - na linha 35;
brincar - na linha 24;
brinque - na linha 95;
brigar - na linha 118;
brasilha - na linha 172;
burguesia - na linha 169;
cafezinho - na linha 31;
caixinha - na linha 16;
cameras - na linha 149;
cansado - na linha 29;
cao - na linha 102;
capitao - na linha 78;
carpinteiro - na linha 45; - na linha 84;
casou - na linha 129;
casar - na linha 83; - na linha 125;
casa - na linha 95; - na linha 123; - na linha 127;
causa - na linha 28;
cegou - na linha 155;
ceilandia - na linha 133;
cem - na linha 46;
central - na linha 40;
certo - na linha 103;
chamou - na linha 132;
chegando - na linha 31; - na linha 127;
chorou - na linha 127;
cidade - na linha 23; - na linha 41; - na linha 43; - na linha 47; - na linha 63; - na linha 69; - na linha 169;
cinco - na linha 164;
circo - na linha 154;
classe - na linha 28; - na linha 88;
coisas - na linha 20; - na linha 52;
colocavam - na linha 16;
com - na linha 12; - na linha 14;
comia - na linha 23;
como - na linha 27;
comecar - na linha 54;
comeco - na linha 44;
comprou - na linha 39;
com - na linha 32; - na linha 41;
cor - na linha 28;
cortar - na linha 45;
conhecia - na linha 49;
comecar - na linha 62;
começou - na linha 70;
como - na linha 60;
com - na linha 66; - na linha 86; - na linha 88;

```
sua - na linha 29; - na linha 28; - na linha 99; - na linha 98; - na linha 99; - na linha 100;
sua - na linha 135;
terror - na linha 13;
tiro - na linha 12;
todos - na linha 8;
todo - na linha 9;
tras - na linha 9;
via - na linha 20;
velhinhas - na linha 16;
ver - na linha 19;
taguatinga - na linha 46;
tal - na linha 111;
tambem - na linha 108; - na linha 137;
tanto - na linha 24; - na linha 26;
televisao - na linha 20; - na linha 140;
tan - na linha 64;
corrido - na linha 76;
tomo - na linha 87; - na linha 126;
tantar - na linha 29;
terror - na linha 26;
ter - na linha 86; - na linha 172;
teu - na linha 162;
todas - na linha 23;
tamar - na linha 21;
tinha - na linha 33; - na linha 77; - na linha 106; - na linha 108;
tiros - na linha 164;
todo - na linha 48;
todas - na linha 66; - na linha 80;
todo - na linha 120; - na linha 144;
toda - na linha 174;
trabalhador - na linha 48;
trabalhando - na linha 106;
trabalhar - na linha 44; - na linha 104;
trabalhava - na linha 56;
traficantes - na linha 66;
traficante - na linha 78; - na linha 112;
traidor - na linha 136;
trazia - na linha 52; - na linha 109;
trouxe - na linha 115;
traidor - na linha 164;
trazia - na linha 157;
tudo - na linha 160; - na linha 162;
viagem - na linha 21;
vida - na linha 27;
uma - na linha 30; - na linha 33; - na linha 61; - na linha 69; - na linha 79; - na linha 81; - na linha 89; - na linha 90; - na linha 90; - na linha 115; - na linha 1
51; - na linha 162;
usar - na linha 117;
vai - na linha 39;
viagem - na linha 33;
visitar - na linha 37;
via - na linha 42;
você - na linha 88;
vão - na linha 74; - na linha 101;
vezia - na linha 172;
velho - na linha 97;
vem - na linha 87; - na linha 162;
ver - na linha 74;
vergonha - na linha 119; - na linha 161;
ver - na linha 125;
ver - na linha 61; - na linha 72; - na linha 128; - na linha 153;
você - na linha 74; - na linha 74;
você - na linha 86; - na linha 98;
voltou - na linha 84;
vou - na linha 74; - na linha 85;
via - na linha 154;
vida - na linha 98; - na linha 99; - na linha 100;
viram - na linha 170;
virou - na linha 154;
viu - na linha 100;
vivera - na linha 152;
você - na linha 99; - na linha 100; - na linha 135; - na linha 136; - na linha 159;
vou - na linha 102; - na linha 125; - na linha 125; - na linha 134;
violencia - na linha 73;
virar - na linha 60;
viviu - na linha 51;
winchester - na linha 115; - na linha 157; - na linha 163;
zona - na linha 47;
C:\Users\Laura\Desktop\avere2\
```

b. Brsail.txt


```
C:\Users\Laura\Desktop\arvore2>gcc mainP.c arvP.c arvP.h -o programaPATRICIA
```

```
C:\Users\Laura\Desktop\arvore2>programaPATRICIA brasil.txt
```

```
antes - na linha 11; - na linha 35;
armaram - na linha 6; - na linha 30;
assim - na linha 23; - na linha 47; - na linha 60; - na linha 68;
brasil - na linha 1; - na linha 20; - na linha 24; - na linha 44; - na linha 48;
cara - na linha 21; - na linha 45; - na linha 58; - na linha 66;
carros - na linha 15;
cartao - na linha 18;
cazuza - na linha 2;
chefe - na linha 17;
cigarro - na linha 13;
compositores - na linha 2;
convencer - na linha 7;
convidaram - na linha 4;
confia - na linha 27;
convencer - na linha 31;
convidaram - na linha 28;
confia - na linha 51; - na linha 64; - na linha 72; - na linha 73;
cores - na linha 40;
credito - na linha 18;
desimportante - na linha 53;
dizer - na linha 43;
droga - na linha 9; - na linha 33;
elegeram - na linha 16;
essa - na linha 9; - na linha 33;
esta - na linha 5;
estacionando - na linha 15;
esta - na linha 29;
fantastico - na linha 37;
festa - na linha 5; - na linha 29;
ficar - na linha 23; - na linha 47; - na linha 60; - na linha 68;
fim - na linha 39;
fiquei - na linha 14;
garota - na linha 37;
gente - na linha 23; - na linha 47; - na linha 60; - na linha 68;
grande - na linha 52;
homens - na linha 6; - na linha 30;
indio - na linha 41;
```



```
.\Users\Laura\Desktop\arvore2>gcc mainP.c arvP.c arvP.h -o programaPATRICIA
```

```
.\Users\Laura\Desktop\arvore2>programaPATRICIA brasil.txt
```

```
antes - na linha 11; - na linha 35;  
arranhar - na linha 6; - na linha 30;  
assin - na linha 23; - na linha 47; - na linha 60; - na linha 68;  
brasil - na linha 1; - na linha 20; - na linha 24; - na linha 44; - na linha 48; - na linha 57; - na linha 61; - na linha 65; - na linha 69; - na linha 74;  
cara - na linha 21; - na linha 45; - na linha 58; - na linha 66;  
carros - na linha 15;  
cartao - na linha 18;  
casuza - na linha 2;  
chefe - na linha 17;  
cigarro - na linha 13;  
compositores - na linha 2;  
convencer - na linha 7;  
convidaram - na linha 4;  
confia - na linha 27;  
convencer - na linha 31;  
convidaram - na linha 28;  
confia - na linha 51; - na linha 64; - na linha 72; - na linha 73;  
cores - na linha 40;  
credito - na linha 18;  
desimportante - na linha 53;  
fizer - na linha 43;  
frega - na linha 9; - na linha 33;  
elegeram - na linha 16;  
essa - na linha 9; - na linha 33;  
esta - na linha 5;  
estacionando - na linha 15;  
esta - na linha 29;  
fantastico - na linha 37;  
festa - na linha 5; - na linha 29;  
ficar - na linha 23; - na linha 47; - na linha 60; - na linha 68;  
fin - na linha 39;  
fiquei - na linha 14;  
garota - na linha 37;  
gente - na linha 23; - na linha 47; - na linha 60; - na linha 68;  
grande - na linha 52;  
homens - na linha 6; - na linha 30;  
indio - na linha 41;  
instante - na linha 54;  
israel - na linha 2;  
malhada - na linha 10; - na linha 34;  
meu - na linha 18; - na linha 39;  
rim - na linha 27; - na linha 51; - na linha 64; - na linha 72; - na linha 73;  
nostra - na linha 21; - na linha 45; - na linha 58; - na linha 66;  
nada - na linha 17;  
nao - na linha 16; - na linha 28; - na linha 36; - na linha 38; - na linha 56; - na linha 56;  
nascer - na linha 35;  
navalha - na linha 19;  
negocio - na linha 25; - na linha 49; - na linha 62; - na linha 70;  
nem - na linha 13;  
nenhum - na linha 54;  
nome - na linha 26; - na linha 50; - na linha 63; - na linha 71;  
nao - na linha 4; - na linha 12;  
nascer - na linha 11;  
ofereceram - na linha 12;  
pagar - na linha 8; - na linha 32;  
paga - na linha 22; - na linha 46; - na linha 59; - na linha 67;  
patria - na linha 52;  
pobre - na linha 5; - na linha 29;  
porta - na linha 14;  
pra - na linha 5; - na linha 7; - na linha 23; - na linha 29; - na linha 31; - na linha 43; - na linha 47; - na linha 60; - na linha 68;  
programada - na linha 42;  
que - na linha 6; - na linha 10;  
quem - na linha 22;  
quero - na linha 22;  
qual - na linha 25; - na linha 49; - na linha 62; - na linha 70;  
que - na linha 30; - na linha 34; - na linha 39;  
quem - na linha 46; - na linha 59; - na linha 67;  
quero - na linha 46; - na linha 59; - na linha 67;  
romero - na linha 2;  
som - na linha 8; - na linha 32;  
vera - na linha 39;  
socio - na linha 26;  
sortearam - na linha 36;
```