

ACH2023 — Algoritmos e Estruturas de Dados  
Exercício Programa I

Igor Nascimento Silva

16 de Dezembro de 2025

# Capítulo 1

## Estrutura

O programa se divide em três arquivos principais: arvore.c, lista.c, 17076671.c. Os arquivos **arvore.c** e **lista.c** contêm as funções padrão de suas respectivas estruturas. A diferença mais notória nessas estruturas é a existência da variável **ocorrencias** do tipo **Ocorrencia**, uma espécie de lista ligada para cuidar de palavras duplicadas na leitura do arquivo. Já o arquivo **17076671.c** é onde “a mágica acontece”, é nele que ocorre a leitura do arquivo, criação do índice e o loop principal do programa e, mesmo que com alguma repetição, a criação de cada índice foi totalmente separada para facilitar a leitura.

Em uma análise simples, o programa **valgrind** aponta não ter vazamentos de memória possíveis nos três cenários que consegui pensar:

1. Arquivo inválido
2. Índice inválido
3. Execução normal

The figure consists of three vertically stacked screenshots of a terminal window. Each screenshot shows the command \$ valgrind ./EP textos/cormen.txt arvore being run. The first two screenshots show errors related to file and index invalidity, while the third shows a successful execution with heap statistics.

```
$ valgrind ./EP textos/cormen.txt arvore
Arquivo: 'textos/cormen.txt'
Tipo de índice: 'arvore'
Erro! Impossível abrir o arquivo de entrada!
==ddd== HEAP SUMMARY:
==ddd==     in use at exit: 0 bytes in 0 blocks
==ddd==   total heap usage: 2 allocs, 2 frees, 1,496 bytes allocated
==ddd== All heap blocks were freed -- no leaks are possible
==ddd== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

```
$ valgrind ./EP textos/texto.txt AVL
Arquivo: 'textos/texto.txt'
Tipo de índice: 'AVL'
Erro! Tipo de índice inválido!
==ddd== HEAP SUMMARY:
==ddd==     in use at exit: 0 bytes in 0 blocks
==ddd==   total heap usage: 2 allocs, 2 frees, 1,496 bytes allocated
==ddd== All heap blocks were freed -- no leaks are possible
==ddd== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

```
$ valgrind ./EP textos/texto.txt arvore
Arquivo: 'textos/texto.txt'
Tipo de índice: 'arvore'
Total de palavras únicas indexadas: 88070
Número de linhas no arquivo: 15
Altura da árvore: 1
> f1m
==ddd== HEAP SUMMARY:
==ddd==     in use at exit: 0 bytes in 0 blocks
==ddd==   total heap usage: 236 allocs, 236 frees, 16,614 bytes allocated
==ddd== All heap blocks were freed -- no leaks are possible
==ddd== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Figura 1.1: Saída nos três casos.

# Capítulo 2

## Resultados

### 2.1 Texto exemplo

O texto exemplo é o trecho apresentado no enunciado do EP. Ele serve como um texto pequeno, mas mostra exemplos esclarecedores. Aqui o uso como indicado no enunciado usando os índices árvore e lista: Em média, as árvores fazem

```
$ ./EP textos/texto.txt arvore
Arquivo: 'textos/texto.txt'
Tipo de indice: 'arvore'
Total de linhas unicas indexadas: 88078
Numero de linhas no arquivo: 13
> busca algoritmo
Existem 4 ocorrências da palavra 'algoritmo' na(s) seguinte(s) linha(s):
88001: Informally, an algorithm is any well-defined computational procedure that takes
88003: as output. An algorithm is thus a sequence of computational steps that transform
88005: inputs according to well-defined rules. In mathematics and computer science
88006: typically, an algorithm is an effective method used for solving a well-specified computational
Numero de comparacoes: 88011
> busca set
Existem 2 ocorrências da palavra 'set' na(s) seguinte(s) linha(s):
88007: some value, or set of values, as input and produces some value, or set of values,
Numero de comparacoes: 88008
> busca quicksort
Palavra 'quicksort' nao encontrada.
Numero de comparacoes: 88012
> busca quicksort
Opcão invalida!
> fim
$ 

$ ./EP textos/texto.txt lista
Arquivo: 'textos/texto.txt'
Tipo de indice: 'lista'
Total de linhas unicas indexadas: 88078
Numero de linhas no arquivo: 13
> busca algoritmo
Existem 4 ocorrências da palavra 'algoritmo' na(s) seguinte(s) linha(s):
88001: Informally, an algorithm is any well-defined computational procedure that takes
88003: as output. An algorithm is thus a sequence of computational steps that transform
88005: inputs according to well-defined rules. In mathematics and computer science
88006: typically, an algorithm is an effective method used for solving a well-specified computational
Numero de comparacoes: 88003
> busca set
Existem 2 ocorrências da palavra 'set' na(s) seguinte(s) linha(s):
88002: some value, or set of values, as input and produces some value, or set of values,
Numero de comparacoes: 88015
> busca quicksort
Palavra 'quicksort' nao encontrada.
Numero de comparacoes: 88078
> busca quicksort
Opcão invalida!
> fim
$ 
```

Figura 2.1: Exemplo mencionado.

menos comparações, mas o arquivo é muito pequeno notar alguma diferença considerável de tempo. A criação do índice é rápida e a busca é mais rápida ainda.

## 2.2 Crepúsculo dos ídolos: ou A filosofia a golpes de martelo

Um texto que serve como uma entrada de tamanho médio. Agora é possível notar uma pequena diferença de tempo na criação do índice, com a lista demorando um pouco mais, mas a busca ainda é muito rápida. O número de comparações na lista já se tornou por volta de 100x o número de comparações na árvore.

```
$ time ./EP textos/pg52263.txt arvore
Arquivo: 'textos/pg52263.txt'
Tipo de indice: 'arvore'
Total de palavras unicas indexadas: 12318
Número de linhas no arquivo: 8442
Altura da arvore: 0003
> busca schopenhauer
Existem 18 ocorrências da palavra 'schopenhauer' no(s) seguinte(s) linha(s):
01537: Schopenhauer, into that principle in which morality appears in its real
02516: Let me point to one single instance. Schopenhauer speaks of beauty with
02508: 'Agn., or -In Dialectics.' Schopenhauer and the honour of
02831: 'The World as Will and Idea,' as Schopenhauer did, one should in the
02652: first place deny Schopenhauer. ... Incidentally, Pessimism, however
03441: [3] A disciple of Schopenhauer who blunted the sharpness of his
03660: regard to the pessimism of the Greeks, as Schopenhauer maintains,
03911: in his 'Inscrutableness of the Right in this respect: by means of
03924: Schopenhauer was hostile to life; that is why he elevated pity to a
03938: that which possessed Schopenhauer (and unfortunately the whole of our
Número de comparações: 00091
> fim
real 0m4.322s
user 0m0.390s
sys 0m0.013s

$ time ./EP textos/pg52263.txt lista
Arquivo: 'textos/pg52263.txt'
Tipo de indice: 'lista'
Total de palavras unicas indexadas: 12318
Número de linhas no arquivo: 8442
> busca schopenhauer
Existem 18 ocorrências da palavra 'schopenhauer' no(s) seguinte(s) linha(s):
01537: Schopenhauer, into that principle in which morality appears in its real
02516: Let me point to one single instance. Schopenhauer speaks of beauty with
02508: 'Agn., or -In Dialectics.' In opposition to Schopenhauer and to the honour of
02831: 'The World as Will and Idea,' as Schopenhauer did, one should in the
02652: first place deny Schopenhauer. ... Incidentally, Pessimism, however
03441: [3] A disciple of Schopenhauer who blunted the sharpness of his
03660: regard to the pessimism of the Greeks, as Schopenhauer maintains,
03911: in his 'Inscrutableness of the Right in this respect: by means of
03924: Schopenhauer was hostile to life; that is why he elevated pity to a
03938: that which possessed Schopenhauer (and unfortunately the whole of our
Número de comparações: 03411
> fim
real 0m6.582s
user 0m2.058s
sys 0m0.021s
```

Figura 2.2: Tempo de criação de índice e busca em cada índice.

## 2.3 Memórias póstumas de Brás Cubas

Outro texto que serve como uma entrada de tamanho médio, porém está aqui para testar palavras com acentuação.<sup>1</sup> Palavras acentuadas são diferentes de palavras não acentuadas (e.g. “está” ≠ “esta”), e tudo funciona normalmente. O número de comparações na lista também é cerca de 100x o número de comparações na árvore.

```
$ time ./EP textos/pg54829.txt arvore
Arquivo: 'textos/pg54829.txt'
Tipo de indice: 'arvore'
Total de palavras unicas indexadas: 13573
Número de linhas no arquivo: 8825
Altura da arvore: 0004
> busca canapé
Existem 6 ocorrências da palavra 'canapé' no(s) seguinte(s) linha(s):
04088: e sentou-se no canapé. Esteve assim alguns instantes, sem me dizer
04046: cortinas do outro, das cadeiras, do tapete, do canapé, de todas essas
04044: do canapé, ou para o interior ouvir os dengueis de D. Plácida.
04088: primeira vez deixou-se cair no canapé, ofegante, escarlate, com os
05213: Virgílio deixou-se cair, no canapé, a rir.—Porque? disse ella.
06123: que esquecesse; ella afastou-se da porta e foi cair no canapé.
Número de comparações: 00045
> fim
real 0m4.212s
user 0m0.396s
sys 0m0.006s

$ time ./EP textos/pg54829.txt lista
Arquivo: 'textos/pg54829.txt'
Tipo de indice: 'lista'
Total de palavras unicas indexadas: 13573
Número de linhas no arquivo: 8825
> busca canapé
Existem 6 ocorrências da palavra 'canapé' no(s) seguinte(s) linha(s):
04088: e sentou-se no canapé. Esteve assim alguns instantes, sem me dizer
04046: cortinas do outro, das cadeiras, do tapete, do canapé, de todas essas
04044: do canapé, ou para o interior ouvir os dengueis de D. Plácida.
04088: primeira vez deixou-se cair no canapé, ofegante, escarlate, com os
05213: Virgílio deixou-se cair, no canapé, a rir.—Porque? disse ella.
06123: que esquecesse; ella afastou-se da porta e foi cair no canapé.
Número de comparações: 07924
> fim
real 0m6.517s
user 0m3.028s
sys 0m0.004s
```

Figura 2.3: Acentos funcionam como você esperaria.

<sup>1</sup>A ideia original era usar um texto em japonês, mas o tratamento de caracteres Unicode e separação de palavras se mostrou muito difícil.

## 2.4 Anna Karenina

Um texto que serve como uma entrada de tamanho grande. A criação de cada índice agora toma tempo considerável e a diferença entre a criação da árvore e a da lista já não é mais marginal. A busca ainda é surpreendentemente rápida para os dois índices. O número de comparações na lista é cerca de 1000x o número de comparações na árvore.

```
time ./EP textos/pg1399.txt arvore
Arquivo: 'textos/pg1399.txt'
Tipo de índice: 'arvore'
Total de palavras unicas indexadas: 21435
Número de linhas no arquivo: 48225
Altura da arvore: 40036
> busca schopenhauer
Existem 2 ocorrências da palavra 'schopenhauer' na(s) seguinte(s) linha(s):
38445: Schopenhauer, the philosophers who gave a non-materialistic explanation
38462: At one time, reading Schopenhauer, he put in place of his will, the
Número de comparações: 00037
> busca algoritmo
Palavra 'algoritmo' não encontrada.
Número de comparações: 00037
> fim

real 0m29.027s
user 0m22.033s
sys 0m0.068s
```

```
time ./EP textos/pg1399.txt lista
Arquivo: 'textos/pg1399.txt'
Tipo de índice: 'lista'
Total de palavras unicas indexadas: 21435
Número de linhas no arquivo: 48225
> busca schopenhauer
Existem 2 ocorrências da palavra 'schopenhauer' na(s) seguinte(s) linha(s):
38445: Schopenhauer, the philosophers who gave a non-materialistic explanation
38462: At one time, reading Schopenhauer, he put in place of his will, the
Número de comparações: 20661
> busca algoritmo
Palavra 'algoritmo' não encontrada.
Número de comparações: 21435
> fim

real 1m3.366s
user 0m56.559s
sys 0m0.092s
```

Figura 2.4: Diferença entre os índices se torna mais notória.

## 2.5 Dicionário

Um exemplo de entrada muito grande. O desempenho da lista é desproporcionalmente pior do que o da árvore a ponto de, no primeiro momento, me questionar se a demora se tratava de um *bug*. Acredito que a explicação se encontra no fato da lista ter de fazer muitas buscas na construção do índice e (quase) todas palavras serem inseridas ao final dela, a árvore lidou muito melhor com o texto em ordem alfabética. O tempo unitário da busca em cada índice continua magnificamente rápido.

```
time ./EP textos/words.txt arvore
Arquivo: 'textos/words.txt'
Tipo de índice: 'arvore'
Total de palavras unicas indexadas: 422987
Número de linhas no arquivo: 466590
Altura da arvore: 63343
> busca zionism
Existem 4 ocorrências da palavra 'zionism' na(s) seguinte(s) linha(s):
200691: Anti-zionism
317795: Pro-zionism
317796: Zionism
465708: Zionism
Número de comparações: 03053
> busca canapé
Palavra 'canapé' não encontrada.
Número de comparações: 03346
> fim

real 0m11.720s
user 0m4.993s
sys 0m0.072s
```

```
time ./EP textos/words.txt lista
Arquivo: 'textos/words.txt'
Tipo de índice: 'lista'
Total de palavras unicas indexadas: 422987
Número de linhas no arquivo: 466590
> busca zionism
Existem 4 ocorrências da palavra 'zionism' na(s) seguinte(s) linha(s):
337795: Pro-zionism
358932: Semi-zionism
465788: Zionism
Número de comparações: 19839
> busca canapé
Palavra 'canapé' não encontrada.
Número de comparações: 422987
> fim

real 3m57.177s
user 3m40.184s
sys 0m3.468s
```

Figura 2.5: Diferença desproporcional entre os índices.

# Capítulo 3

## Conclusão

O desempenho de cada índice depende mais do formato do arquivo de entrada do que o tamanho dele. O maior gargalo do programa se encontra na criação do índice, algo que só se torna útil caso o usuário queira fazer sucessivas buscas no arquivo de entrada e faz o programa ficar extremamente mais lento caso contrário.

Usei o dicionário como exemplo de resultado porque acreditava se tratar do pior caso para os dois índices, mas essa ideia intuitiva se mostrou falsa. Uma lista sequencial ao invés de uma lista ligada poderia se beneficiar da busca no caso de uma entrada em ordem alfabética, porém é um caso muito específico e a inserção ainda seria lenta.

**Fato curioso:** O programa usou, no máximo, 75.9MB em ambos os índices (A árvore usa um pouco mais de memória). Achei curioso porque, dado o tamanho da entrada, pensava que seria muito mais. A título de comparação, o leitor de pdf em que estou editando o relatório, que é conhecido por ser “leve” e “minimalista” (**Zathura**), usa cerca de 172.5MB e o editor de texto (**Neovim**) usa 30.3MB.