

Relatório Trabalhos Práticos AEDS3

Felipe Carvalho, João Vitor Baldo

Ciência da Computação – PUC Minas

Resumo

Neste relatório fala sobre como foram feitos os trabalhos práticos para a matéria Algoritmos e Estruturas de Dados 3. Para o trabalho prático 1 era necessário implementar CRUD Sequencial e Ordenação Externa. Para o trabalho prático 2 era necessário implementar indexação utilizando Árvore B, Hashing Estendido e Lista Invertida. Para o trabalho prático 3 era necessário fazer a implementação dos algoritmos Huffman, LZW e dois algoritmos de casamento de padrões Para o último trabalho prático era necessário implementar dois algoritmos de criptografia.

Trabalho Prático 1 - Criação da base de dados, Manipulação de Arquivo Sequencial e Ordenação Externa

No primeiro trabalho prático da disciplina, desenvolvemos uma aplicação para manipular e organizar informações dos personagens dos filmes da saga Harry Potter.

O trabalho começou com a implementação de uma classe para realizar a leitura de um arquivo CSV contendo os dados dos personagens, como identificador, nome, data de nascimento, e outros atributos. Essas informações são armazenadas em um arquivo binário com o seguinte formato:

- Um número inteiro no início do arquivo para registrar o último ID inserido.
- Uma lápide de 1 byte que indica se o registro é válido (1 para válido e 0 para excluído).
- Um inteiro representando o tamanho (em bytes) do registro do personagem.
- Os dados do personagem são armazenados em formato estruturado, incluindo inteiros, strings e datas (representadas como valores long).

Foi implementado também operações básicas de CRUD (Create, Read, Update e Delete), permitindo a manipulação completa dos registros de personagens.

Por fim, foi desenvolvido um algoritmo de ordenação externa para reorganizar os registros no arquivo binário. Este algoritmo foi projetado para lidar com grandes volumes de dados, que excedem a capacidade de memória principal, utilizando estratégias de divisão, ordenação em blocos menores e intercalamento.

Trabalho Prático 2: Manipulação de Arquivo Indexado com Árvore B, Hash e Lista Invertida

No segundo trabalho prático, foi implementado a inclusão de estruturas de dados avançadas para otimizar as operações de busca e manipulação no arquivo de dados. Foram implementadas três abordagens principais: Árvores B, Hashing Estendido e Listas Invertidas.

-Árvore B de Ordem 8 com Arquivo de Índice

Foi implementada uma Árvore B de ordem 8, acompanhada de um arquivo de índice para armazenar os pares "ID do personagem, posição no arquivo de dados".

A Árvore B foi projetada para realizar buscas eficientes no arquivo binário de dados, utilizando o índice para localizar rapidamente a posição de um personagem específico. Sempre que havia alterações no arquivo de dados, como inserções, exclusões ou atualizações, o arquivo de índice era automaticamente atualizado para refletir essas mudanças. Garantindo um desempenho otimizado mesmo em grandes volumes de registros.

-Hashing Estendido com Arquivo de Índice

Também foi implementado um Hashing Estendido utilizando um arquivo de índice que mapeava o ID do personagem e posição no arquivo de dados.

A função hash utilizada foi $h(k) = k \bmod 2^p$, onde p representa a profundidade do diretório. Cada bucket podia armazenar até X registros, sendo X equivalente a 2% do tamanho inicial da base (404 registros). Sempre que um bucket atingia sua capacidade máxima, acontecia um split no diretório, adaptando a estrutura para comportar novos registros. O índice era atualizado em tempo real sempre que o arquivo de dados sofria alterações, garantindo a

consistência entre os dois arquivos. O objetivo do Hashing Estendido era fornecer uma alternativa eficiente para buscas rápidas no arquivo de dados.

-Lista Invertida

E por último, foi implementado um sistema de Listas Invertidas para permitir buscas mais flexíveis no arquivo binário de dados.

As listas invertidas armazenavam informações de índices secundários, associando atributos não primários, como ano de nascimento ou casa do personagem, às posições dos registros no arquivo de dados. Sempre que um registro era inserido, alterado ou deletado no arquivo de dados, as listas invertidas eram atualizadas para refletir essas mudanças. O sistema foi projetado para permitir buscas que utilizassem múltiplas listas invertidas simultaneamente, possibilitando pesquisas complexas que combinavam diferentes critérios, como, por exemplo, personagens de uma determinada casa e com uma data de nascimento específica.

Trabalho Prático 3: Compactação e Casamento de Padrões

No terceiro trabalho prático, foi implementado algoritmos de compactação de dados e casamento de padrões. O objetivo foi tornar o sistema mais eficiente no armazenamento e processamento de informações, utilizando algoritmos consagrados para cada tarefa.

-Compactação do Arquivo Binário

Foi adicionada ao programa a capacidade de compactar o arquivo binário de dados, permitindo ao usuário escolher quando realizar esse processo. Para isso, foram utilizados dois algoritmos de compactação:

- **Huffman:** Baseado em árvores binárias, esse algoritmo atribui códigos menores a caracteres mais frequentes, maximizando a eficiência da compactação.

- **LZW:** Utiliza um dicionário dinâmico para identificar padrões repetitivos no arquivo e reduzir seu tamanho de forma eficiente.

Após a compactação, o programa oferece ao usuário a opção de descompactar o arquivo.

Quando essa operação era realizada, o arquivo descompactado substitui o arquivo de dados e passa a ser utilizado como o novo arquivo de dados pelo programa, garantindo que as operações subsequentes fossem realizadas sobre os dados descompactados.

O sistema também apresentava ao usuário métricas importantes, como:

- O tempo de execução dos algoritmos Huffman e LZW durante a compactação.
- A porcentagem de perda após a descompactação comparando o tamanho com o antigo arquivo de dados.

-Casamento de Padrões

Foi implementada ao Read a funcionalidade de busca de padrões nos registros dos personagens, utilizando os algoritmos KMP e Rabin-Karp.

O usuário fornece o ID do personagem e o padrão que deseja encontrar no registro do personagem e o programa realiza a busca utilizando os dois algoritmos, no final o programa mostra para o usuário se o padrão desejado foi encontrado ou não nos dados do personagem.

Trabalho Prático 4: Criptografia

O quarto trabalho prático teve objetivo a implementação de técnicas de criptografia, adicionando ao programa funcionalidades para proteger os dados armazenados e processados. Foram aplicados dois algoritmos amplamente utilizados no campo da segurança da informação: RSA e Cifra Sequencial.

O programa foi configurado para executar as operações de criptografia e descriptografia quando o usuário digitar "Encriptar" no console. Quando ativado, ele realiza a criptografia e a descriptografia do arquivo de dados gerando no total 4 arquivos, sendo 2 criptografados pelos algoritmos RSA e cifra por substituição e 2 arquivos descriptografados.

-RSA

Foi implementado o algoritmo RSA, que utiliza a criptografia assimétrica para proteger os dados.

O programa utiliza uma chave pública para fazer a criptografia dos dados e uma chave privada que necessita ser mantida em sigilo para a fazer descriptografia dos dados.

Ao digitar "Encriptar" o programa realiza a criptografia dos dados e em seguida realiza a descriptografia dos dados.

-Cifra por Substituição

Foi implementada também a Cifra por Substituição, uma técnica de criptografia simétrica simples que substitui cada caractere por outro de acordo com uma regra definida.

Para a criptografia o programa aplica uma chave de substituição predefinida, onde o byte do arquivo é substituído pelo valor na tabela.

Para a descriptografia o programa faz o processo inverso gerando o arquivo igual o arquivo de dados.

Testes

-Arquivo CSV de origem

```
1 id,name,alternate_names,house,ancestry,species,patronus,hogwartsStaff,hogwartsStudent,actorName,alive,alternate_actors,dateOfBirth,yearOfBirth,eyeColour,gender,hairColour,wizard
2 9e3f7ce4-b9a7-4244-b799-daec5cf1d4a8;Harry Potter;["The Boy Who Lived", "The Chosen One", "Undesirable No. 1", "Potty"];Gryffindor;half-blood;human;stag;FALSE;VERDADEIRO;Daniel Radcliffe;VERDADEIRO
3 4c7e6819-a91e-45b2-a454-f931e4a7ccc3;Hermione Granger;["Herm", "Know-it-all", "Miss Grant", "Herm-on-nimby"];Gryffindor;muggleborn;human;otter;FALSE;VERDADEIRO;Emma Watson;VERDADEIRO;[[19-09-1997]]
4 c3b1f9a5-b870-48b4-b0d0-95b091eeb390;Ron Weasley;["Dragonfire", "Ronald", "Liddle Ronnikins", "Ronnie", "Weezy", "Hon-Mon", "Ronnie", "Ronnie"];Gryffindor;pure-blood;human;Jack Russell terrier
5 af95bd8a-dfae-45bb-bc69-53386bd34129;Draco Malfoy;[[Slytherin;pure-blood;human;Jack Russell terrier;FALSE;VERDADEIRO;Tom Felton;VERDADEIRO;[[05-06-1980]]];grey;male;blonde;VERDADEIRO
6 ca3827f0-375a-4891-aa45-f5e8a5bd2255;Minerva McGonagall;[[Gryffindor;half-blood;human;tabby cat;VERDADEIRO;FALSE;Dame Maggie Smith;VERDADEIRO;[[04-10-1925]]];grey;female;black;VERDADEIRO
7 d5c4daa1-c79e-426a-a0b8-fb0bf1fb0816;Cedric Diggory;[[Hufflepuff;half-blood;human;FALSO;VERDADEIRO;Roberts Partridge;FALSE;[[04-10-1925]]];grey;male;brown;VERDADEIRO
8 8f9a4a00-5d7c-4a1e-ad12-4564ecd3b74c;Cho Chang;[[Ravenclaw;half-blood;human;swan;FALSO;VERDADEIRO;Katie Leung;VERDADEIRO;[[07-04-1979]]];brown;female;black;VERDADEIRO
9 3569d265-bd27-44d8-88e8-b2fbb0a84374;Severus Snape;["Half-Blood Prince", "Snivellus", "Sev"];Slytherin;half-blood;human;doe;VERDADEIRO;FALSE;Alan Rickman;FALSE;[[09-01-1960]]];black;male;black
10 36bfef0e-e0bb-4d11-b60e-d1ef617a777a;Rubeus Hagrid;["Hagger"];Gryffindor;half-blood;half-giant;VERDADEIRO;FALSE;Robbie Coltrane;VERDADEIRO;[[06-12-1928]]];black;male;black
11 3bdc6c51-b401-4f44-a6e4-d1ff32221c5;Neville Longbottom;[[Gryffindor;pure-blood;human;Non-Corporal;FALSO;VERDADEIRO;Matthew Lewis;VERDADEIRO;[[30-07-1980]]];black;male;blonde;VERDADEIRO
12 861c4cde-2f0f-479e-8d8f-9402e74b2573;Luna Lovegood;["Loony Lovegood"];Ravenclaw;pure-blood;human;hare;FALSE;VERDADEIRO;Evanna Lynch;VERDADEIRO;[[13-02-1981]]];silver;female;blonde;VERDADEIRO
13 1cd0dc64-e1a9-4379-9cfd-1a767ba1bb1;Ginny Weasley;[[Gryffindor;pure-blood;human;horse;FALSO;VERDADEIRO;Bonnie Wright;VERDADEIRO;[[11-08-1981]]];brown;female;red;VERDADEIRO
14 2cfd2d0e-5d1e-4dc5-8837-7a07c7e2f2f5;Sirius Black;["Padfoot", "Snuffles"];Gryffindor;pure-blood;human;hare;FALSO;FALSO;Gary Oldman;FALSO;["James Walters", "Rohan Gotobed"];03-11-1959;1959;grey;ma
15 ba5f0953-9d6c-4d70-43a0-4391a1f872e4;Remus Lupin;["Professor Lupin", "Moony", "Remus John Lupin"];Gryffindor;half-blood;werewolf;wolf;VERDADEIRO;FALSE;David Thewlis;FALSO;[[10-03-1960]]];green
16 d925874-e800-4eb4-9fd0-4d0fed15634;Arthur Weasley;[[Gryffindor;pure-blood;human;weasel;FALSO;FALSO;Mark Williams;VERDADEIRO;[[06-02-1950]]];blue;male;red;VERDADEIRO
17 6a6b1960-febd-418d-b684-e581c159459b;Bellatrix Lestrange;["Bella"];Slytherin;pure-blood;human;FALSO;FALSO;Helena Bonham Carter;FALSO;[[06-02-1950]]];brown;female;black;VERDADEIRO
18 cf4802c8-ae18-4ae1-4524-49d212d05939;Lord Voldemort;["Tom Riddle", "Tom Marvolo Riddle", "You-Know-Who", "He Who Must Not Be Named", "the Dark Lord", "the Heir of Slytherin"];Slytherin;half-blood
19 2f8675cd-5505-4c8e-a54e-579e73bf4174;Horace Slughorn;[[Slytherin;pure-blood;human;VERDADEIRO;FALSO;Jim Broadbent;VERDADEIRO;[[31-12-1926]]];green;male;blonde;VERDADEIRO
20 106cf68d-a86a-415e-ad75-6ad9a4ae24e4;Kingsley Shacklebolt;[[Slytherin;pure-blood;human;lynx;FALSO;FALSO;George Harris;VERDADEIRO;[[31-12-1926]]];brown;male;brown;VERDADEIRO
21 d58e7249-19d1-40bd-e43f-1da0a97f68aa;Polyurus Underidge;[[Slytherin;half-blood;human;persian cat;VERDADEIRO;FALSO;Penelope Staunton;VERDADEIRO;[[31-12-1926]]];brown;female;grey;VERDADEIRO
22 41403619-70eb-4a0c-b70a-d5c4ae20e602;Lucius Malfoy;[[Slytherin;pure-blood;human;FALSO;FALSO;Jason Isaacs;VERDADEIRO;[[31-12-1926]]];grey;male;blonde;VERDADEIRO
23 0a4f9eb45-d843-4e29-47d3-6bd49e0d7f85;Vincent Crabbe;[[Slytherin;pure-blood;human;FALSO;VERDADEIRO;Jamie Maylett;FALSO;[[31-12-1926]]];black;male;black;VERDADEIRO
24 a31ddc78-af12-4978-929c-3c1a0ab0a33e;Gregory Goyle;[[Slytherin;pure-blood;human;FALSO;VERDADEIRO;Josh Herdman;VERDADEIRO;[[31-12-1926]]];black;male;brown;VERDADEIRO
25 b0a20914-85bd-46fc-baed-033c565e138b;Mrs Norris;[[Slytherin;pure-blood;cat;VERDADEIRO;FALSO;Nastase, Alnott and Toney the cat;VERDADEIRO;[[31-12-1926]]];yellow;female;brown;FALSO
26 2b02cf0b-0440-4a57-a030-df75a40cb090;Argus Filch;[[Slytherin;squib;human;VERDADEIRO;FALSO;David Bradley;VERDADEIRO;[[31-12-1926]]];yellow;male;grey;FALSO
27 b4ffe09b-8796-42ad-a6d9-faf1a369a7e;Vernon Dursley;[[Slytherin;muggle;human;FALSO;FALSO;Richard Griffiths;VERDADEIRO;[[31-12-1926]]];yellow;male;grey;FALSO
28 fedc24df-56d9-495e-bada-ea770d0957e0;Petunia Dursley;[[Slytherin;muggle;human;FALSO;FALSO;Iona Shaw;VERDADEIRO;[[31-12-1926]]];yellow;female;blonde;FALSO
29 7d34c46e-680e-474c-a976-b1e0997637e0;Madley Dursley;[[Slytherin;muggle;human;FALSO;FALSO;Hurry Holling;VERDADEIRO;[[23-6-1900]]];blue;male;blond;FALSO
30 41cd0bbe-e943-431b-bbde-bb2cad3491a1;Lily Potter;[[Gryffindor;muggleborn;human;FALSO;FALSO;Geraldine Somerville;FALSO;[[30-01-1960]]];green;female;blond;VERDADEIRO
31 34eb6182-00cf-4c95-ae73-7d234066d18;James Potter;["Prongs"];Gryffindor;pure-blood;human;stag;FALSO;FALSO;Adrian Rawlins;FALSO;[[27-03-1960]]];green;male;black;VERDADEIRO
32 b415c867-1e4f-455e-b19d-748662ac2cca;Albus Dumbledore;["Professor Dumbledore"];Gryffindor;half-blood;human;Phoenix;VERDADEIRO;FALSO;Richard Harris;FALSO;["Richard Gombos"];27-03-1960;1961;blue;ma
33 e0457467-d10a-4893-af40-104960b2b18a;Nadav Pomfrey;["Poppy Pomfrey"];Gryffindor;half-blood;human;VERDADEIRO;FALSO;Senna Jones;VERDADEIRO;[[27-03-1960]]];blue;female;silver;VERDADEIRO
34 63c6cfff-a1a4-4caf-a5ac-e2197c12c1be;Mrs Figg;["Arabella Doreen Figg"];Gryffindor;squib;human;FALSO;FALSO;Kathryn Hunter;VERDADEIRO;[[27-03-1960]]];blue;female;silver;FALSO
35 162b3473-aaab-46ae-98d0-725e6c0eb084;Nargus Dursley;["Marjorie Eileen Dursley", "Aunt Marge", "Auntie Marge", "Marge"];Gryffindor;muggle;human;FALSO;FALSO;Pam Ferris;VERDADEIRO;[[27-03-1960]]];blue;male;silver;FALSO
36 d70f60ae-810a-4230-b0d0-c852c0811f0;Yemmy;[[Gryffindor;muggle;human;FALSO;FALSO;VERDADEIRO;[[27-03-1960]]];blue;female;silver;FALSO
37 57fa4f38-af1f-4e71-800d-49a738b2a610;Piers Polkiss;[[Gryffindor;muggle;human;FALSO;FALSO;Jason Boyd;VERDADEIRO;[[27-03-1960]]];blue;male;silver;FALSO
38 a18f59a-89cf-4afd-bf87-b3273900d011;Dennis;[[Gryffindor;muggle;human;FALSO;FALSO;Christopher Rithin;VERDADEIRO;[[27-03-1960]]];blue;male;silver;FALSO
39 e43d1ee-427c-465e-0a9f-037442b376d3;Nolcote;[[Gryffindor;muggle;human;FALSO;FALSO;Richard Nockin;VERDADEIRO;[[27-03-1960]]];blue;male;silver;FALSO
40 71a95495-134d-4a7f-8a73-5f3a95970845;Gordon;[[Gryffindor;muggle;human;FALSO;FALSO;VERDADEIRO;[[27-03-1960]]];blue;male;silver;FALSO
```

Todos os algoritmos feitos têm este csv como base e realizam operações nos registros levando em conta os tipos de seus atributos.

-Arquivo Binário

O arquivo binário gerado que tem a seguinte formatação:

- Um número inteiro no início do arquivo para registrar o último ID inserido.
- Uma lápide de 1 byte que indica se o registro é válido (1 para válido e 0 para excluído).
- Um inteiro representando o tamanho (em bytes) do registro do personagem.
- Os dados do personagem são armazenados em formato estruturado, incluindo inteiros, strings e datas (representadas como valores long).

characters.bin		Decoded Text
00000000	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F \$ 9
00000010	65 33 66 37 63 65 34 2D 62 39 61 37 2D 34 32 34	e 3 f 7 c e 4 - b 9 a 7 - 4 2 4
00000020	34 2D 62 37 30 39 2D 64 61 65 35 63 31 66 31 64	4 - b 7 0 9 - d a e 5 c 1 f 1 d
00000030	34 61 38 00 0C 48 61 72 72 79 20 50 6F 74 74 65	4 a 8 . . H a r r y P o t t e
00000040	72 00 3D 5B 54 68 65 20 42 6F 79 20 57 68 6F 20	r . = [T h e B o y W h o
00000050	4C 69 76 65 64 2C 20 54 68 65 20 43 68 6F 73 65	L i v e d , T h e C h o s e
00000060	6E 20 4F 6E 65 2C 20 55 6E 64 65 73 69 72 61 62	n O n e , U n d e s i r a b
00000070	6C 65 20 4E 6F 2E 20 31 2C 20 50 6F 74 74 79 5D	l e N o . 1 , P o t t y]
00000080	00 0A 47 72 79 66 66 69 6E 64 6F 72 00 0A 68 61	. . G r y f f i n d o r . . h a
00000090	6C 66 2D 62 6C 6F 6F 64 00 05 68 75 6D 61 6E 00	l f - b l o o d . . h u m a n .
000000A0	04 73 74 61 67 00 05 66 61 6C 73 65 00 04 74 72	. s t a g . . f a l s e . . t r
000000B0	75 65 00 10 44 61 6E 69 65 6C 20 52 61 64 63 6C	u e . . D a n i e l R a d c l
000000C0	69 66 66 65 00 04 74 72 75 65 00 00 00 4D BB A1	i f f e . . t r u e . . M . .
000000D0	6B 80 00 05 67 72 65 65 6E 00 04 6D 61 6C 65 00	k . . g r e e n . . m a l e .
000000E0	05 62 6C 61 63 6B 00 04 74 72 75 65 00 00 07 BC	. b l a c k . . t r u e
000000F0	00 00 00 00 DC 00 00 00 01 00 24 34 63 37 65 36 \$ 4 c 7 e 6
00000100	38 31 39 2D 61 39 31 61 2D 34 35 62 32 2D 61 34	8 1 9 - a 9 1 a - 4 5 b 2 - a 4
00000110	35 34 2D 66 39 33 31 65 34 61 37 63 63 65 33 00	5 4 - f 9 3 1 e 4 a 7 c c e 3 .
00000120	10 48 65 72 6D 69 6F 6E 65 20 47 72 61 6E 67 65	. H e r m i o n e G r a n g e
00000130	72 00 30 5B 48 65 72 6D 79 2C 20 4B 6E 6F 77 2D	r . 0 [H e r m y , K n o w -
00000140	69 74 2D 61 6C 6C 2C 20 4D 69 73 73 20 47 72 61	i t - a l l , M i s s G r a
00000150	6E 74 2C 20 48 65 72 6D 2D 6F 77 6E 2D 6E 69 6E	n t , H e r m - o w n - n i n
00000160	6E 79 5D 00 0A 47 72 79 66 66 69 6E 64 6F 72 00	n y] . . G r y f f i n d o r .
00000170	0A 6D 75 67 67 6C 65 62 6F 72 6E 00 05 68 75 6D	. m u g g l e b o r n . . h u m
00000180	61 6E 00 05 6F 74 74 65 72 00 05 66 61 6C 73 65	a n . . o t t e r . . f a l s e
00000190	00 04 74 72 75 65 00 0B 45 6D 6D 61 20 57 61 74	. . t r u e . . E m m a W a t
000001A0	73 6F 6E 00 04 74 72 75 65 00 00 00 47 60 47 DB	s o n . . t r u e . . G ` G .
000001B0	80 00 05 62 72 6F 77 6E 00 06 66 65 6D 61 6C 65	. . . b r o w n . . f e m a l e
000001C0	00 05 62 72 6F 77 6E 00 04 74 72 75 65 00 00 07	. . b r o w n . . t r u e . . .
000001D0	BB 00 00 00 01 06 00 00 00 02 00 24 63 33 62 31 \$ c 3 b 1
000001E0	66 39 61 35 2D 62 38 37 62 2D 34 38 62 66 2D 62	f 9 a 5 - b 8 7 b - 4 8 b f - b
000001F0	30 30 64 2D 39 35 62 30 39 33 65 61 36 33 39 30	0 0 d - 9 5 b 0 9 3 e a 6 3 9 0
00000200	00 0B 52 6F 6E 20 57 65 61 73 6C 65 79 00 54 5B	. . R o n W e a s l e y . T [

-LZW

Após fazer a compactação utilizando o algoritmo LZW, o programa gera uma arquivo com os dados compactados

TP3 > charactersLZWCompressao1.bin	
	Decoded Text
00000000 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	
00000000 00 01 09 EA 00 00 00 00 00 00 00 00 00 00 00 01	
00000010 00 00 00 93 00 00 01 00 00 00 01 00 00 00 00 E7	
00000020 00 00 01 04 00 00 01 00 00 00 24 00 00 00 39	\$. . . 9
00000030 00 00 00 65 00 00 00 33 00 00 00 66 00 00 00 37	e . . . 3 . . . f . . . 7
00000040 00 00 00 63 00 00 00 65 00 00 00 34 00 00 00 2D	c . . . e . . . 4 . . . -
00000050 00 00 00 62 00 00 00 39 00 00 00 61 00 00 00 37	b . . . 9 . . . a . . . 7
00000060 00 00 00 2D 00 00 00 34 00 00 00 32 00 00 00 34	- . . . 4 . . . 2 . . . 4
00000070 00 00 01 11 00 00 00 62 00 00 00 37 00 00 00 30	b . . . 7 . . . 0
00000080 00 00 00 39 00 00 00 2D 00 00 00 64 00 00 00 61	9 . . . - . . . d . . . a
00000090 00 00 00 65 00 00 00 35 00 00 00 63 00 00 00 31	e . . . 5 . . . c . . . 1
000000A0 00 00 00 66 00 00 00 31 00 00 00 64 00 00 00 34	f . . . 1 . . . d . . . 4
000000B0 00 00 00 61 00 00 00 38 00 00 00 00 00 00 00 0C	a . . . 8
000000C0 00 00 00 48 00 00 00 61 00 00 00 72 00 00 00 72	H . . . a . . . r . . . r
000000D0 00 00 00 79 00 00 00 20 00 00 00 50 00 00 00 6F	y P . . . o
000000E0 00 00 00 74 00 00 00 74 00 00 00 65 00 00 00 72	t . . . t . . . e . . . r
000000F0 00 00 00 00 00 00 00 3D 00 00 00 5B 00 00 00 54 = . . . [. . . T
00000100 00 00 00 68 00 00 00 65 00 00 00 20 00 00 00 42	h . . . e B
00000110 00 00 00 6F 00 00 01 33 00 00 00 57 00 00 00 68	o . . . 3 . . . W . . . h
00000120 00 00 00 6F 00 00 00 20 00 00 00 4C 00 00 00 69	o L . . . i
00000130 00 00 00 76 00 00 00 65 00 00 00 64 00 00 00 2C	v . . . e . . . d . . . ,
00000140 00 00 00 20 00 00 01 3E 00 00 01 40 00 00 00 43	. . . > . . . @ . . . C
00000150 00 00 01 46 00 00 00 73 00 00 00 65 00 00 00 6E	F . . . s . . . e . . . n
00000160 00 00 00 20 00 00 00 4F 00 00 00 6E 00 00 00 65 O . . . n . . . e
00000170 00 00 01 4E 00 00 00 55 00 00 00 6E 00 00 00 64	N . . . U . . . n . . . d
00000180 00 00 00 65 00 00 00 73 00 00 00 69 00 00 00 72	e . . . s . . . i . . . r
00000190 00 00 00 61 00 00 00 62 00 00 00 6C 00 00 01 40	a . . . b . . . l . . . @
000001A0 00 00 00 4E 00 00 00 6F 00 00 00 2E 00 00 00 20	N . . . o
000001B0 00 00 00 31 00 00 01 4E 00 00 01 35 00 00 01 37	1 . . . N . . . 5 . . . 7
000001C0 00 00 00 79 00 00 00 5D 00 00 00 00 00 00 00 0A	y . . .]
000001D0 00 00 00 47 00 00 01 32 00 00 00 66 00 00 00 66	G . . . 2 . . . f . . . f
000001E0 00 00 00 69 00 00 01 5D 00 00 00 6F 00 00 01 3A	i . . .] . . . o . . . :
000001F0 00 00 00 0A 00 00 00 68 00 00 00 61 00 00 00 6C h . . . a . . . l
00000200 00 00 00 66 00 00 01 12 00 00 00 6C 00 00 00 6F	f l . . . o

-Indexação pelos atributos “casa” e “YearOfBirth”

```
❏ arquivoCasa.txt
1 Slytherin: [918, 1659, 3193, 3475, 3648, 3830, 4017, 4187, 4359, 4528, 4717, 4881, 5054, 5225, 5393, 12683, 13539, 13715, 14545, 14947, 15120, 17027, 17537, 17712, 17884, 18047, 18225, 18409, 185
2 Hufflepuff: [1283, 8525, 8683, 8841, 9010, 9193, 9358, 9514, 9675, 9852, 10007, 11471, 11628, 11805, 11980, 12865, 15312, 15487, 15649, 15812, 24020, 24208, 33716, 33874, 34030, 34188, 34821, 366
3 Ravenclaw: [1453, 2242, 12147, 12328, 14056, 14218, 14375, 16002, 16186, 16357, 16521, 16682, 22816, 22979, 23143, 23307, 23470, 23653, 24389, 24590, 24762, 24921, 25097, 25272, 25430, 25593, 257
4 Gryffindor: [240, 465, 732, 1188, 1861, 2051, 2418, 2608, 2825, 3001, 5573, 5756, 5959, 6143, 6329, 6554, 6785, 6871, 7038, 7203, 7352, 7513, 7674, 7834, 7990, 8168, 8326, 10206, 10414, 10582, 10
5
```

```
❏ arquivoIdade.txt
1 1954: [4187, 4359, 4528, 4717, 4881, 5054, 5225, 9675, 9852, 10007, 10206, 10414, 10582, 10752, 15312, 15487, 15649, 15812, 16002, 16186, 16357, 20725, 20897, 21075, 21262, 21417, 21589, 21751, 2
2 1925: [1180, 7038, 12505, 18047, 23470, 28999, 34030, 39369, 44681, 49884, 55069, 60143, 65466]
3 1926: [3475, 3648, 3830, 4017, 9010, 9193, 9358, 9514, 14545, 14775, 14947, 15120, 20039, 20187, 20352, 20546, 25593, 25765, 25986, 26164, 30883, 31043, 31198, 31356, 36033, 36211, 36482, 36580,
4 1959: [2600, 8326, 13806, 19388, 24921, 30224, 35386, 40717, 45971, 51214, 56342, 61508, 66815]
5 1960: [1659, 2825, 5573, 5756, 7513, 8525, 11110, 11298, 13064, 14056, 16682, 16859, 18567, 19543, 22113, 22271, 24020, 25097, 27693, 27858, 29457, 30392, 32742, 32901, 34509, 35561, 38064, 38215
6 1928: [1861, 7674, 13226, 18719, 24208, 29611, 34665, 40007, 45322, 50543, 55715, 60835, 66152]
7 1977: [1283, 7203, 12683, 18225, 23653, 29144, 34188, 39543, 44838, 50042, 55235, 60321, 65616]
8 1881: [5959, 6143, 11471, 11628, 17027, 17204, 22456, 22614, 28007, 28166, 33058, 33236, 38390, 38561, 43729, 43889, 48860, 49032, 54093, 54251, 59160, 59311, 64481, 64643]
9 1979: [465, 1453, 6554, 7352, 11000, 12865, 17537, 18409, 22979, 23830, 28533, 29290, 33553, 34350, 38909, 39742, 44704, 44998, 49307, 50202, 54574, 55391, 59637, 60484, 64944, 65777]
10 1980: [240, 732, 918, 2051, 5393, 6329, 6705, 6871, 7834, 10920, 11805, 12147, 12328, 13382, 16521, 17371, 17712, 17884, 18881, 21916, 22816, 23143, 23307, 24389, 27523, 28364, 28699, 28850, 2970
11 1981: [2242, 2418, 7990, 8168, 13539, 13715, 19001, 19225, 24500, 24762, 29022, 30074, 34995, 35191, 40406, 40550, 45657, 45819, 50877, 51049, 56023, 56178, 61165, 61342, 66464, 66652]
12 1950: [3001, 8683, 14218, 19709, 25272, 30543, 35716, 41122, 46282, 51540, 56641, 61852, 67190]
13 1951: [3193, 8041, 14375, 19069, 25430, 30723, 35081, 41292, 46438, 51712, 56780, 62021, 67300]
```

Conclusão

Os trabalhos práticos realizados ao longo deste semestre foram muito importantes para colocar em prática os conceitos que aprendemos em sala de aula. Conseguimos ver na prática o

impacto de escolher algoritmos eficazes e bem estruturados no desempenho e no funcionamento do programa.

Ao longo do semestre, aprendemos diferentes algoritmos de manipulação de arquivos indexados, compactação, casamento de padrões e criptografia e os trabalhos práticos foram importantes para ver na prática o funcionamento desses algoritmos.