



# Pontifícia Universidade Católica de Minas Gerais

Instituto de Ciências Exatas e Informática  
Algoritmos e Estruturas de Dados III | Prof. Felipe Soares

## Lista de Exercícios 1

1 ponto - felipesoares@pucminas.br

- A lista deve ser feita **individualmente** ou em grupos de no **máximo 2 alunos**.
- **Penalidade por atraso:** a cada dia corrido de atraso, a nota será penalizada em 0,5 ponto.
- **Penalidade por cópia:** trabalhos iguais não são aceitos (nota 0).

**Questões em azul:** opcionais (importantes para o estudo, mas não contabilizam pontos para a lista)

**Questões em preto:** contabilizam pontos para a lista

## Parte teórica (0,5 ponto)

### Conceitos Básicos / Organização de Campos e Registros / Manipulação de Arquivos / Arq. Sequenciais

1 - Qual é a menor quantidade de bytes lida ou escrita em um disco rígido por operação? Justifique.

2 - Qual é o significado do termo cilindro na estrutura de um disco rígido?

3 - EXPLIQUE por que a gravação de um arquivo em um disco rígido (HDD) é feita cilindro a cilindro ao invés de prato a prato.

4 - EXPLIQUE os componentes do tempo de acesso em discos rígidos (HDD).

5 - DESCREVA e JUSTIFIQUE três características de um bom identificador de entidades em arquivos indexados.

6 - Um arquivo binário é apenas uma sequência de bytes. Dessa forma, quaisquer valores a serem armazenados em um arquivo também devem ser representados como sequências de bytes. Considerando isso, EXPLIQUE como a data e a hora de um evento (por exemplo, data e hora de uma compra em uma loja virtual) podem ser armazenadas em um arquivo usando a menor quantidade possível de bytes. EXEMPLIFIQUE.

7 - Em um arquivo indexado, os registros excluídos não são removidos imediatamente do arquivo, mas são apenas marcados como excluídos, por meio de um campo lógico chamado lápide. Como a exclusão não é, portanto, efetivada imediatamente (apenas durante uma reordenação), esses registros excluídos ficam ocupando espaço desnecessariamente no arquivo. Torna-se oportuna, então, a criação de uma estratégia que permita o reaproveitamento desses espaços quando fizermos inserções de novos registros. Para isso, precisamos, em primeiro lugar, saber onde estão esses espaços e quais são os seus tamanhos.

ELABORE uma estratégia eficiente para localização dos espaços dos registros marcados para exclusão em um arquivo indexado com registros de tamanho variável.



## Pontifícia Universidade Católica de Minas Gerais

### Instituto de Ciências Exatas e Informática

#### Algoritmos e Estruturas de Dados III | Prof. Felipe Soares

**8** - DESCREVA uma estrutura de registro de tamanho variável, que permita o armazenamento dos registros abaixo. Evite qualquer uso desnecessário ou redundante de bytes.

ID: 44

Nome: José de Almeida Santos

Data de nascimento: 04/12/1998

Gênero: M

Cidade: Belo Horizonte

UF: MG

ID: 33

Nome: Carla Maria de Oliveira

Data de nascimento: 23/03/1990

Gênero: F

Cidade: Rio de Janeiro

UF: RJ

ID: 37

Nome: Pedro Roberto de Souza

Idade: 12/06/2003

Gênero: M

Cidade: São Paulo

UF: SP

**9** - EXPLIQUE duas vantagens do armazenamento de datas usando um tipo int ou long ao invés do tipo string. Justifique cada uma dessas vantagens. Explique quando seria necessário usar int e quando seria necessário usar long.

**10** - É possível se criar registros de tamanho fixo com campos de tamanho variável? JUSTIFIQUE sua resposta demonstrando como.

**11** - DESCREVA uma estrutura de registro de tamanho variável, que permita o armazenamento dos registros abaixo. Esses registros descrevem tipos de infração de trânsito. Evite qualquer uso desnecessário ou redundante de bytes.

Código: 501-00

Infração: Dirigir veículo sem CNH

Pontos: 7

Valor: R\$ 574,62

Código: 518-51

Infração: Não usar cinto de segurança

Pontos: 5

Valor: R\$ 127,69

Código: 523-11

Infração: Atirar objetos do veículo

Pontos: 4

Valor: R\$ 85,13



## Pontifícia Universidade Católica de Minas Gerais

### Instituto de Ciências Exatas e Informática

#### Algoritmos e Estruturas de Dados III | Prof. Felipe Soares

**12** - Vários sites na Web permitem que usuários cadastrados armazenem seus dados ou ações no próprio site. Por exemplo, o site IMDB (Internet Movie Data Base) permite que um usuário dê notas aos filmes que assistiu. Essas notas ficam armazenadas no site permanentemente e são usadas para determinação do perfil do usuário e, mais tarde, para se criar uma lista de recomendações de filmes.

Para poder realizar ações assim, é importante que o usuário já tenha feito o login, geralmente informando o seu email e a sua senha.

Considerando isso, EXPLIQUE quais devem ser os campos de um registro USUÁRIO cujos objetivos são permitir o login e o registro das notas dadas a filmes no site. JUSTIFIQUE a necessidade de cada atributo e informe também qual deve ser o seu tipo e, quando for o caso, se é de tamanho fixo ou variável.

**13** - Considere um sistema que armazena, em arquivos, strings de tamanho variável usando a codificação UTF-8. Cada string, nesse sistema, é armazenada com o seu próprio indicador de tamanho, cujo tamanho deve ser considerado.

Quantos bytes serão usados no arquivo por esse sistema para armazenar a string EDUCAÇÃO?

**14** - Um arquivo sequencial é necessariamente ordenado? JUSTIFIQUE a sua resposta.

**15** - A forma tradicional de exclusão em um arquivo sequencial é apenas marcar o registro como excluído, por meio de um campo chamado lápide. EXPLIQUE uma vantagem e uma desvantagem dessa forma de exclusão quando comparada à forma básica que seria remover imediatamente o registro, liberando o espaço ocupado por ele (isto é, o arquivo reduziria de tamanho).

**16** - EXPLIQUE o que é o cabeçalho de um arquivo de dados? CITE um exemplo de informação que ele pode conter.

**17** - Qual dos tipos de dados é a melhor opção para uso como identificador de clientes em um sistema de gestão empresarial, considerando que esse sistema tem, atualmente, 350 clientes e uma expectativa de crescimento de 150 clientes por ano?

**18** - Para que servem os campos lápide em um arquivo?

**19** - O espaço de registros marcados como excluídos pode ser reaproveitado em novas inclusões, desde que o arquivo não seja ordenado ou, caso seja ordenado, que a sua ordem seja preservada.

No entanto, ainda é importante considerar o tamanho do novo registro e o tamanho da área que se deseja reaproveitar. Se o novo registro for exatamente do mesmo tamanho do espaço, então não há problemas. Se for maior, esse espaço não poderá ser usado.

Se, porém, o registro for menor que o espaço disponível, haverá alguma sobra de bytes nesse espaço. Qual deve ser a ação nesse caso?

**20** - Considere uma alteração de um registro em um arquivo sequencial ordenado que resultou na redução do tamanho desse registro. EXPLIQUE como deve ser gerenciado o espaço que deixou de ser necessário.



## Ordenação Externa

1 - Logo após a fase inicial de distribuição, qual é a quantidade máxima de elementos ordenados que podemos assegurar em cada segmento (e ainda antes de iniciarmos as intercalações)?

Considere uma intercalação balanceada de 2 caminhos e uma capacidade de ordenação em memória principal de apenas 5 elementos.

2 - Quantas etapas de intercalação (passadas pelos arquivos) serão necessárias para se ordenar completamente um arquivo que contenha 5.000 entidades, sabendo que a capacidade de ordenação em memória principal é de 25 entidades e que há três arquivos temporários em cada conjunto (3 caminhos)?

Nesse cálculo, não inclua a fase inicial de distribuição das entidades pelos caminhos.

3 - Qual a vantagem de se trabalhar com segmentos de tamanho variável em uma intercalação balanceada?

4 - Qual o principal benefício do uso de uma estrutura como o heap de mínimo na fase de distribuição nos oferece?

5 - Suponha que você tem um arquivo com 10.000 entidades que precisar ser ordenado. Considere que você consegue ordenar, em memória principal, apenas 300 entidades, usando algum algoritmo como Quicksort ou Mergesort, e que seu algoritmo de ordenação externa realiza a intercalação balanceada básica de 2 caminhos.

Quantos segmentos ordenados existirão em cada um dos arquivos temporários, logo após a distribuição inicial (e ainda antes de qualquer intercalação)?

6 - Considerando a capacidade de ordenação em memória de 3 registros, faça a intercalação balanceada com segmentos de tamanho variável e 2 caminhos para os registros, cujas chaves são listadas a seguir. Considere um algoritmo que faça a intercalação usando  $n+1$  arquivos temporários, sendo  $n$  o total de caminhos. Mostre todos os passos da intercalação.

51	11	19	12	8	88	78	97	92	24	18	2	89	76	39	75	69	27	67	71
----	----	----	----	---	----	----	----	----	----	----	---	----	----	----	----	----	----	----	----

7 - Considerando a capacidade de ordenação em memória de 4 registros, FAÇA a intercalação polifásica de 2 caminhos para os registros, cujas chaves (não exclusivas) são listadas a seguir. APRESENTE todos os passos da intercalação.

A	C	N	I	O	A	E	C	I	A	O	C	E	D	E	F	L	A	E	Q
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

8 - EXPLIQUE como o uso de um heap de mínimo pode aumentar a eficiência da intercalação balanceada.



## Pontifícia Universidade Católica de Minas Gerais

### Instituto de Ciências Exatas e Informática

#### Algoritmos e Estruturas de Dados III | Prof. Felipe Soares

**9** - Considerando a capacidade de ordenação em memória de 3 registros e usando heap como estrutura para ajudar na ordenação. FAÇA a intercalação balanceada de 2 caminhos com segmentos de tamanho variável para os registros, cujas chaves (não exclusivas) são listadas a seguir. APRESENTE todos os passos da intercalação.

A	L	G	O	R	I	T	M	O	S	E	E	S	T	R	U	T	U	R	A	S
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**10** - Considerando a capacidade de ordenação em memória de 3 registros, FAÇA a intercalação balanceada de 2 caminhos para os registros, cujas chaves são listadas a seguir. APRESENTE todos os passos da intercalação.

20	11	23	03	19	97	66	62	40	55	48	31	29	79	52	61
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

**11** - Considerando a capacidade de ordenação em memória de 3 registros, FAÇA a intercalação polifásica de 3 caminhos com segmentos de tamanho variável para os registros, cujas chaves são listadas a seguir. APRESENTE todos os passos da intercalação.

32	69	15	20	18	45	17	37	29	54	13	66	9	36	44	31	87	78	49	50
----	----	----	----	----	----	----	----	----	----	----	----	---	----	----	----	----	----	----	----

**12** - Considerando a capacidade de ordenação em memória de 3 registros, FAÇA a intercalação balanceada de 2 caminhos para os registros, usando  $n+1$  arquivos temporários, sendo  $n$  o total de caminhos, cujas chaves (não exclusivas) são listadas a seguir. APRESENTE todos os passos da intercalação.

C	O	N	V	E	R	G	E	N	C	I	A	D	I	G	I	T	A	L
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

## Arq. Indexados / Árvores B, B+ e B\*

**1** - Os arquivos indexados são mais eficientes que os arquivos sequenciais em operações baseadas em acesso aleatório. DESCREVA as sequências de passos para se fazer buscas aleatórias em cada um desses tipos de arquivos e, em seguida, EXPLIQUE por que os arquivos indexados são mais eficientes nessa operação.

**2** - EXEMPLIFIQUE a estrutura de um arquivo indexado. Para isso, você deve representar um conjunto de entidades quaisquer indicando o endereço (byte inicial) de cada entidade no arquivo. Você pode usar uma tabela para apresentar esses dados. Você também deve apresentar os dados que serão armazenados no índice (considere um índice secundário, denso e direto). Novamente, você pode apresentar o índice na forma de uma tabela.



## Pontifícia Universidade Católica de Minas Gerais

### Instituto de Ciências Exatas e Informática

#### Algoritmos e Estruturas de Dados III | Prof. Felipe Soares

**3** - DESCREVA a sequência de passos para se fazer uma exclusão de uma entidade em um arquivo indexado. O identificador dessa entidade será informado pelo usuário.

**4** - EXPLIQUE uma vantagem e uma desvantagem dos arquivos indexados sobre os arquivos sequenciais.

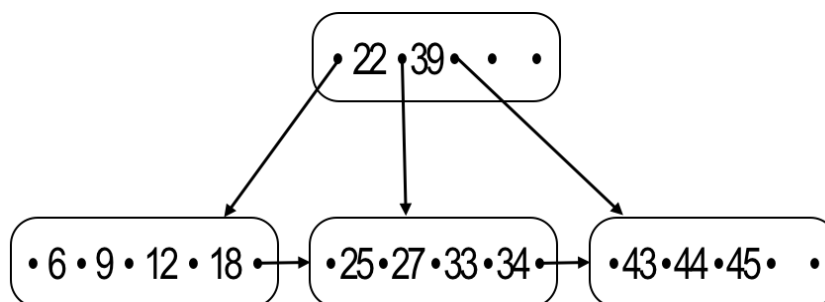
**5** - Em um arquivo indexado, não há uma ordem física específica dos registros no arquivo, pois o acesso aos mesmos é aleatório. Se os registros tiverem tamanho variável, a operação de alteração pode ser feita de uma forma bem simples: basta marcar o registro anterior para exclusão (usando o campo lápide) e reinserir o registro atualizado no arquivo. Isso, porém, resulta em espaços de diversos tamanhos desperdiçados no meio do arquivo.

DESCREVA uma estratégia eficiente para se aproveitar esses espaços desperdiçados quando for realizada uma operação de inserção de registro. Sua estratégia não deve depender de um segundo arquivo, mesmo que temporário.

**6** - EXPLIQUE o que é um índice secundário, direto e denso. Em seguida, EXEMPLIFIQUE esse tipo de índice, por meio do desenho de um arquivo de dados e do seu índice, ambos na forma de tabelas.

**7** - Uma árvore B de ordem 3 e com 3 níveis de altura (raiz + 2 níveis) possui no mínimo quantas chaves?

**8** - Considere a seguinte árvore B+ de ordem 5:



- a) Qual será a árvore resultante se acrescentarmos um elemento de chave 20 a essa árvore?
- b) Qual será a árvore resultante de removermos os elementos 12, 9 e 6?
- c) Qual será essa mesma árvore usando a estrutura B? Repita as letras a e b para a árvore B.
- d) Como é o arquivo para a árvore B e B+ de ambas as árvores iniciais?

**10** - Uma árvore B de ordem 4 é construída do zero e recebe 10 inserções de chaves. Qual será o número máximo de divisões de páginas que pode ocorrer durante essas inserções?

**11** - Crie uma árvore B\* de ordem 5 vazia e acrescente a ela as chaves: 45 37 54 23 51 63 42 60 13. Em seguida, remova as chaves: 13 51 54 37 45 23 63. Redesenhe a árvore a cada inserção ou remoção.

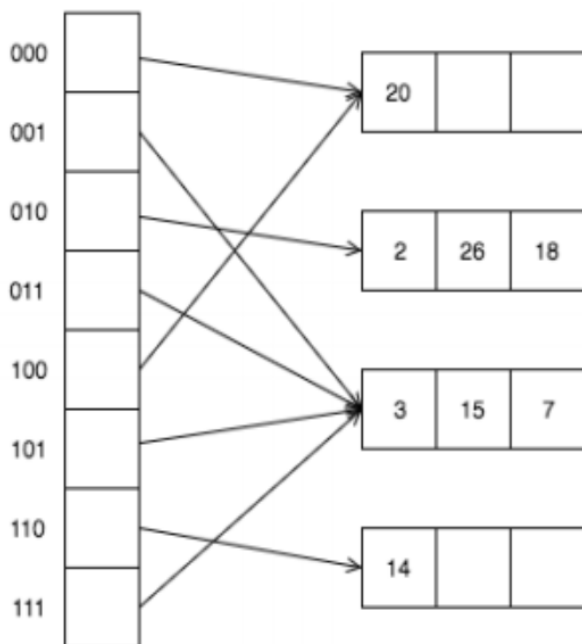


## Hashing

1 - Qual é a vantagem do uso de buckets nas tabelas hash em disco?

2 - O que pode ser calculado a partir da profundidade do diretório de uma tabela hash extensível?

3 - Considere a seguinte tabela hash extensível:



Suponha que façamos a inserção das chaves 30, 21, 12, 42 e 31, nesta ordem. Que chave provocará o primeiro aumento de profundidade do diretório?

4 - Considere a tabela hash extensível abaixo, que possui a função hash  $h(k) = k \bmod 2p$ , em que  $p$  é o número de bits (profundidade) usado no diretório.

Acrescente a essa tabela hash os pares de chave e endereço (7;372), (17;115) e (6;278). Redesenhe tanto o diretório quanto o conjunto de cestos sempre que um novo cesto for criado durante as inserções, bem como a solução final.

Diretório			Buckets							
	p = 1	posição	prof.	n	chave	end.	chave	end.	chave	end.
0	0	→ 0	p'=1	3	8	217	20	567	30	996
1	1	→ 1	p'=1	2	1	305	15	697		



## Pontifícia Universidade Católica de Minas Gerais

### Instituto de Ciências Exatas e Informática

#### Algoritmos e Estruturas de Dados III | Prof. Felipe Soares

**5** - Criar uma tabela hash dinâmica com a seguinte função hash:  $h(x) = x \bmod 2^p$ , em que  $p$  é a profundidade do diretório dessa tabela. Considere que, inicialmente o diretório tem profundidade 1 e que cada bucket é capaz de armazenar 3 registros.

Insira, nessa tabela, os registros com as seguintes chaves:

20, 15, 16, 19, 10, 30, 13, 8, 4, 5

## Todo conteúdo

**1** - Suponha que as seguintes chaves foram inseridas em uma estrutura de pesquisa: 10 / 03 / 14 / 18 / 20 / 08 / 06 / 01 / 12 / 22 / 07 / 16 / 13 / 19 (nesta ordem) e em seguida foram removidas as chaves 18 e 07 (nesta ordem).

- Pede-se que se represente a estrutura resultante caso seja utilizada Árvore B de ordem 03.
- Pede-se que se represente a estrutura resultante caso seja utilizada Árvore B<sup>+</sup> de ordem 04.
- Pede-se que se represente a estrutura resultante caso seja utilizada Árvore B<sup>\*</sup> de ordem 02.
- Hashing Estendido, que possui a função hash  $h(k) = k \bmod 2^p$ , em que  $p$  é o número de bits (profundidade) usado no diretório, cada bucket podendo armazenar até 03 registros.
- Para cada um dos exercícios anteriores, mostre como ficou o arquivo final.

**2** - Suponha que registros com as seguintes chaves foram inseridos em uma estrutura de pesquisa na memória secundária: 11 / 04 / 15 / 19 / 21 / 09 / 07 / 02 / 13 / 23 / 08 / 17 / 14 / 20 (nesta ordem) e que, em seguida, foram removidos os registros de chaves 19 e 08 (nesta ordem).

- Pede-se que se represente a estrutura resultante caso seja utilizada a Árvore B de ordem 03.
- Pede-se que se represente a estrutura resultante caso seja utilizada Árvore B<sup>+</sup> de ordem 04.
- Pede-se que se represente a estrutura resultante caso seja utilizada Árvore B<sup>\*</sup> de ordem 02.
- Hashing Estendido, que possui a função hash  $h(k) = k \bmod 2^p$ , em que  $p$  é o número de bits (profundidade) usado no diretório, cada bucket podendo armazenar até 04 registros.
- Para cada um dos exercícios anteriores, mostre como ficou o arquivo final.





# Pontifícia Universidade Católica de Minas Gerais

## Instituto de Ciências Exatas e Informática

Algoritmos e Estruturas de Dados III | Prof. Felipe Soares

### Parte Implementação (0,5 ponto)

1 - implementar um sistema responsável por realizar operações de CRUD (create, read, update e delete) em um arquivo sequencial. Para a realização dessas operações, você usará o contexto de contas bancárias, com os campos:

- **idConta** (deve ser incremental à medida que novos registros forem adicionados) (int)
- **nomePessoa** (string de tamanho variável)
- **email** (1 ou mais) (strings de tamanhos variáveis com indicador de quantidade)
- **nomeUsuario** (string de tamanho variável)
- **senha** (string de tamanho variável)
- **cpf** (string de tamanho fixo igual a 11)
- **cidade** (string de tamanho variável)
- **transferenciasRealizadas** (int)
- **saldoConta** (float)

Observações:

- Deve-se utilizar cabeçalho para armazenar o último valor de id utilizado;
- Deve-se usar lápide - Byte que indica se o registro é válido ou se é um registro excluído;
- Deve-se usar Indicador de tamanho do registro - Número inteiro que indica o tamanho do vetor de bytes.

O sistema deverá oferecer uma tela inicial (com uso pelo terminal) com um menu com as seguintes opções:

- Criar uma conta bancária -> essa escolha deve, a partir da leitura dos dados da conta bancária pelo terminal (nomePessoa, cpf, cidade, emails (1 ou mais), nomeUsuario e senha), criar uma nova conta no arquivo (saldoConta e transferenciasRealizadas devem ser iniciados com "valor informado pelo terminal" e 0, respectivamente). Obs: o nomeUsuario deve ser único para cada cliente.
- realizar uma transferência -> essa escolha deve atualizar dados em duas contas no arquivo.
  - Para isso, é necessário permitir ao usuário cadastrar uma operação de transferência, ou seja, informar duas contas (uma para debitar e outra para creditar o valor) e o valor da transferência. Assim, a conta para debitar deve ter uma redução em saldoConta do valor da transferência, enquanto que a conta para creditar deve receber um acréscimo. Então, o programa deve atualizar o campo saldoConta e o campo transferenciasRealizadas (em +1) das duas contas.
- Ler um registro (id) -> esse método deve receber um id como parâmetro, percorrer o arquivo e retornar os dados do id informado.
- Atualizar um registro -> esse método deve receber novas informações sobre um objeto e atualizar os valores dele no arquivo. Observe duas possibilidades que podem acontecer:
  - O registro mantém seu tamanho - Nenhum problema aqui. Basta atualizar os dados no próprio local.
  - O registro aumenta ou diminui de tamanho - O registro anterior deve ser apagado (por meio da marcação lápide) e o novo registro deve ser escrito no fim do arquivo.
- Deletar um registro (id) -> esse método deve receber um id como parâmetro, percorrer o arquivo e colocar uma marcação (lápide) no registro que será considerado deletado.



# **Pontifícia Universidade Católica de Minas Gerais**

## **Instituto de Ciências Exatas e Informática**

### **Algoritmos e Estruturas de Dados III | Prof. Felipe Soares**

- Realizar a ordenação do arquivo. Para isso, deve ser realizada a implementação da ordenação externa, considerando a memória principal com limitação de “m” registros (parametrizável) e usando “n” caminhos (parametrizável). Implemente os seguintes algoritmos:
  - a) Intercalação balanceada comum
  - b) Intercalação balanceada com blocos de tamanho variável
  - c) Intercalação balanceada com seleção por substituição
  - d) Intercalação balanceada usando  $n+1$  arquivos
  - e) Intercalação Polifásica