

AEDs III

Primeira lista de exercícios

1. O planejamento do tamanho de um registro em um arquivo de dados deve levar em consideração o tamanho do setor do disco rígido? Justifique sua resposta.
2. Considerando as características de um disco rígido, explique por que a leitura de registros sequenciais em disco é mais rápida do que a leitura de registros aleatórios.
3. Quantos bytes são necessários para se armazenar um caráter usando a codificação UTF-8? Justifique sua resposta.
4. Explique qual é a estrutura adequada para se armazenar os seguintes registros, considerando que os registros são de tamanho variável, com indicador de tamanho. Em seguida, calcule quantos bytes são necessários para se armazenar os três registros.

Id: 1	Nome: TV	Cor: Preta	Garantia extra: Não	Preço: R\$ 3.428,00
Id: 2	Nome: Refrigerador	Cor: Branca	Garantia extra: Sim	Preço: R\$ 4.399,90
Id: 3	Nome: Fogão	Cor: Inox	Garantia extra: Sim	Preço: R\$ 928,00

5. Descreva como deve ser a rotina de alteração em um arquivo sequencial com registros de tamanho variável, considerando: (a) que a alteração reduzirá o número de bytes do registro e (b) que a alteração aumentará o número de bytes do registro.
6. Considerando a capacidade de ordenação em memória de 4 registros, faça a intercalação balanceada usando 3 caminhos, com segmentos de tamanho fixo, para os registros cujas chaves são listadas a seguir. Mostre todos os passos da intercalação.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
67	94	90	76	21	73	39	76	44	69	17	88	92	51	68	1	64	34	29	26	54	16	69	84

25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
61	88	61	19	28	64	47	83	51	87	9	55	15	12	21	94	43	31	20	25	43	59	78	45

49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
36	62	54	15	72	62	47	97	34	53	44	7	34	77	51	38	4	88	10	9	92	52	59	73

7. Considerando a capacidade de ordenação em memória de 4 registros, faça a intercalação balanceada usando 3 caminhos, com segmentos de tamanho variável, para os registros cujas chaves são listadas a seguir. Mostre todos os passos da intercalação.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
L	P	O	N	Q	G	J	E	W	L	K	P	Q	Z	T	W	C	R	D	T	W	Y	V	D

25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
C	L	Q	K	K	R	H	S	D	K	N	M	L	C	M	L	W	J	L	Q	P	Y	O	Y

49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
V	Y	P	N	A	N	N	X	Q	V	A	X	L	J	V	K	R	V	G	F	I	Q	F	H

8. Se atualizarmos um campo que não é chave (ex: endereço, idade, etc.) de um registro de tamanho variável de um arquivo indexado, haverá alguma situação em que o índice também deverá ser atualizado? Justifique.
9. Cite duas vantagens e duas desvantagens dos arquivos indexados sobre os arquivos sequenciais.
10. Escreva a função `create()` em Java para um CRUD de filmes. Para tanto, considere as seguintes características do arquivo de filmes:
 - O cabeçalho do arquivo contém o último ID usado e um contador de registros válidos (filmes não excluídos);
 - Não há uma ordem específica para os registros no arquivo;
 - Cada registro no arquivo é composto por um campo `lápide`, um indicador de tamanho do vetor de bytes do tipo `int` e o próprio vetor de bytes que contém os dados de um filme;
 - A função `create()` receberá um objeto `Filme`. Ela deve retornar o ID do novo filme e pode devolver as exceções disparadas;
 - O objeto da classe `Filme` possui os métodos `toByteArray()` e `setID()` como visto em sala.

GABARITO

Questão 1

Sim. O setor é a menor unidade de dados de um disco rígido. Assim, em todas as operações feitas em disco, setores inteiros são lidos ou escritos. Se não houver uma correspondência entre o tamanho do setor e o tamanho do registro, seja na forma de um setor que comporta vários registros ou de um registro que ocupa vários setores, então é possível que mais setores sejam lidos/escritos do que o necessário, caso um planejamento melhor tivesse sido feito.

Questão 2

A leitura de registros sequenciais tira proveito de duas características do disco rígido. A primeira é o fato de reduzir o número de acessos a posições diferentes do disco, que consome tempo extra (tempo de acesso). Quando o próximo registro está logo após o registro atual, então não é necessário um novo acesso.

A segunda característica é que quando fazemos uma leitura em disco, os próximos setores já são armazenados no buffer do disco, de tal forma que a recuperação do próximo registro, se estiver entre esses setores, será bem mais rápida.

Questão 3

A codificação UTF-8 é de tamanho variável. Assim, podem ser usados 1, 2, 3 ou 4 bytes.

A quantidade de bytes está associada à frequência com que o símbolo é usado. Por exemplo, os caracteres da tabela ASCII (até a posição 127) são armazenados com 1 bytes apenas. Caracteres acentuados dos idiomas latinos consomem 2 bytes. Símbolos especiais, raramente usados, geralmente consomem 4 bytes.

Questão 4

Indicador de tamanho do registro = int (4 bytes)

Campo ID = int (4 bytes)

Campo Nome = string de tamanho variável, usando indicador de tamanho de 2 bytes

Campo Cor = string de tamanho variável, usando indicador de tamanho de 2 bytes.

(Uma variação seria usar uma tabela de cores com índice numérico.)

Campo Garantia Estendida = boolean (1 byte)

Campo Preço = float (4 bytes)

Tamanho total = 24 (primeiro registro) + 35 (segundo registro) + 27 (terceiro registro) = 86 bytes

(obs.: no terceiro registro, o caráter 'ã' consome 2 bytes)

Questão 5

- a) Se o registro alterado for menor que o registro original, então é possível que ele seja mantido no mesmo espaço do registro original, com um pouco de "lixo" no seu fim. Esse lixo, porém, não pode ser ignorado no processamento sequencial, pois, dessa forma, perderíamos a referência ao início do próximo registro. Há duas alternativas possíveis para considerá-lo: incluir os bytes desse lixo no indicador de tamanho do registro ou criar um campo lixo no registro que, para os demais, será uma sequência de bytes vazia.*
- b) A solução quando o registro alterado é maior que o registro original, depende de saber se o arquivo é ordenado. Se o arquivo não for ordenado, então basta marcar o registro original como excluído (por meio do campo lápide) e incluir o registro alterado no fim do arquivo. Se, porém, o arquivo for ordenado, então a solução é criar o espaço para esse novo registro, deslocando os demais alguns bytes para a frente. A forma mais simples de fazer isso é com o apoio de um arquivo auxiliar, para o qual são copiados todos os registros posteriores ao alterado. Após a efetiva alteração do registro, os demais são copiados de volta do arquivo temporário para o arquivo de dados. Uma alternativa seria colocar o novo registro na área de extensão (marcando o original como excluído), caso o arquivo a possua.*

Questão 6

1:	67	76	90	94	1	51	68	92	19	61	61	88	12	15	21	94	15	36	54	62	34	38	51	77												
2:	21	39	73	76	26	29	34	64	28	47	64	83	20	25	31	43	47	62	72	97	4	9	10	88												
3:	17	44	69	88	16	54	69	84	9	51	55	87	43	45	59	78	7	34	44	53	52	59	73	92												
4:	17	21	39	44	67	69	73	76	76	88	90	94	12	15	20	21	25	31	43	43	45	59	78	94												
5:	1	16	26	29	34	51	54	64	68	69	84	92	7	15	34	36	44	47	53	54	62	62	72	97												
6:	9	19	28	47	51	55	61	61	64	83	87	88	4	9	10	34	38	51	52	59	73	77	88	92												
1:	1	9	16	17	19	21	26	28	29	34	39	44	47	51	51	54	55	61	61	64	64	67	68	69	69	73	76	76	83	84	87	88	88	90	92	94
2:	4	7	9	10	12	15	15	20	21	25	31	34	34	36	38	43	43	44	45	47	51	52	53	54	59	59	62	62	72	73	77	78	88	92	94	97
3:																																				
4:	1	4	7	9	9	10	12	15	15	16	17	19	20	21	21	25	26	28	29	31	34	34	34	36	38	39	43	43	44	44	45	47	47	51	51	51
	52	53	54	54	55	59	59	61	61	62	62	64	64	67	68	69	69	72	73	73	76	76	77	78	83	84	87	88	88	88	90	92	92	94	94	97
5:																																				
6:																																				

Questão 7

1:	L	N	O	P	Q	T	W	Z	C	K	L	Q	C	L	L	M	N	P	V	Y	J	K	L	V												
2:	E	G	J	Q	C	D	R	T	H	K	R	S	J	L	Q	W	A	N	N	X	F	G	R	V												
3:	K	L	P	W	D	V	W	Y	D	K	M	N	O	P	Y	Y	A	Q	V	X	F	H	I	Q												
4:	E	G	J	K	L	L	N	O	P	P	Q	Q	T	W	W	Z	A	J	J	K	L	L	Q	Q	V	V	W	X								
5:	C	C	D	D	K	L	Q	R	T	V	W	Y	A	F	H	I	N	N	Q	X																
6:	C	D	H	K	K	L	L	M	M	N	N	O	P	P	R	S	V	Y	Y	Y	F	G	R	V												
1:	C	C	C	D	D	D	E	G	H	J	K	K	K	K	L	L	L	L	L	M	M	N	N	N	O	O	P	P	P	P	Q	Q	Q	R	R	S
	T	T	V	V	W	W	Y	Y	Y	Y	Z																									
2:	A	A	F	F	G	H	I	J	J	K	L	L	N	N	Q	Q	Q	R	V	V	V	V	W	X	X											
3:																																				
4:	A	A	C	C	C	D	D	D	E	F	F	G	G	H	H	I	J	J	J	K	K	K	K	K	L	L	L	L	L	L	L	M	M	N	N	N
	N	N	O	O	P	P	P	P	Q	Q	Q	Q	Q	Q	R	R	R	S	T	T	V	V	V	V	V	W	W	W	W	X	X	Y	Y	Y	Y	Z
5:																																				
6:																																				

Questão 8

A alteração de campos que não são chave apenas elimina a necessidade de um índice de chave primária. Assim, se esse campo for usado em um índice secundário, então apenas esse índice precisará ser atualizado.

Se, na alteração, o registro mudar de tamanho, é importante considerar se esse registro (ou qualquer outro) precisará ser reposicionado. Isso geralmente acontece quando a alteração resulta em aumento do tamanho do registro. Nesse caso, qualquer índice direto, especialmente o baseado na chave primária, precisará ser alterado.

Questão 9

VANTAGENS:

- *Os arquivos indexados, por meio de vários índices, permitem a busca rápida por diferentes campos.*
- *Uma alteração em um registro, não implicará em movimentação de outros registros.*

DESVANTAGENS:

- *O processamento sequencial em arquivos indexados é mais lento, pois os registros serão localizados no arquivo de dados individualmente (a partir da ordenação dada pelo índice).*
- *É necessário manter vários arquivos: um de dados e outro para cada índice.*

Questão 10

```
public int create(Filme novoFilme) throws Exception {  
  
    // Lê o cabeçalho  
    arq.seek(0);  
    int id = arq.readInt() + 1;  
    int qtdeRegistros = arq.readInt();  
  
    // Grava o novo registro  
    novoFilme.setID(id);  
    arq.seek(arq.length());  
    byte[] b = novoFilme.toByteArray();  
    int tam = (int) b.length;  
    arq.write(' ');  
    arq.writeInt(tam);  
    arq.write(b);  
  
    // Atualiza o cabeçalho  
    arq.seek(0);  
    arq.writeInt(id);  
    arq.writeInt(qtdeRegistros+1);  
  
    return id;  
}
```