



1. Considere o seguinte esquema:

Empregados(matr, nome, ender, salário, lotação).

A tabela *Empregados* contém os dados dos funcionários de uma empresa Z. O atributo *matr* representa a chave primária da relação. O fator de página de *Empregados* é 100. O parâmetro  $V(\text{matr}, \text{Empregado})$  tem valor igual a  $10^6$ . Foi definido um índice primário denso sobre o atributo *salário*. O fator de seletividade do atributo *salário* em *Empregados* é 400. Suponha que a seguinte consulta foi submetida ao SGBD:

*Select \* from Empregados where salário = 200*

Calcule a estimativa de custo para a estratégia mais eficiente dessa consulta. Considere, que no momento da consulta, o melhor caso para a árvore  $B^+$  (de ordem 4) correspondente ao índice sobre *salário*.

2. Considere duas relações não-ordenadas *r* e *s*, para quais não foram definidos índices. Qual o mais baixo custo para computar a junção entre *r* e *s*, assumindo uma área de buffer infinita? Qual a quantidade de buffer necessária para o algoritmo de junção neste cenário?
3. Construa a **árvore  $B^+$**  de ordem 4 com a seguinte ordem de inserção 4,5,3,9,6,12,13,14,2,15,17

4. Considere o seguinte esquema:

Fornecedor(cod\_for, nome, endereço)

Histórico(matr\_vend, cod\_item, dt\_hora\_venda, qtde\_venda, pr\_venda)

Item(cod\_item, referência, qtde\_estoque, localização, cod\_for)

Os atributos sublinhados representam os atributos chave primária das relações do esquema acima. O atributo *cod\_for* é chave estrangeira, referenciando a chave primária de *Fornecedor*. O atributo *cod\_item* é chave estrangeira, referenciando a chave primária de *Item*. Considere  $V(\text{cod\_for}, \text{Fornecedor})=300$ ,  $V((\text{matr\_vend}, \text{cod\_item}, \text{dt\_hora\_venda}), \text{Histórico})=2,9 \times 10^4$ ,  $V(\text{cod\_item}, \text{Item})=4,8 \times 10^4$ ,  $f_{\text{FORNECEDOR}}=50$ ,  $f_{\text{HISTÓRICO}}=30$  e  $f_{\text{ITEM}}=120$ . Considere ainda que o fator de página do resultado de *Fornecedor*  $\bowtie$  *Item* é 20 e de *Histórico*  $\bowtie$  *Item* é 40. Suponha que a seguinte consulta foi submetida ao SGBD:

*select \* from Fornecedor f inner join Item i inner join  
histórico h on i.cod\_item=h.cod\_item on f.cod\_for=i.cod\_for*

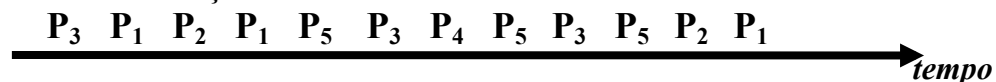
**Calcule a estimativa de custo** para o processador de consultas executar a consulta acima, considerando as seguintes estratégias:

(a) block nested loop join; (b) merge-join

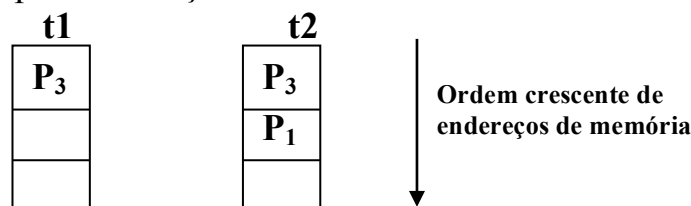
**Indique qual a estratégia mais eficiente para executar a operação de junção definida acima e construa o plano de acesso para a consulta.**



5. Uma variação da política de alocação de páginas em buffer LRU, é denominada de SLRU. Nessa política, há uma tabela de páginas denominada de tabela de páginas SLRU. Para cada página  $P_i$  na tabela, existe um bit de referência  $R_i$  associado com a página. Se há uma solicitação de acesso a  $P_i$ , e  $P_i$  não está na área de *buffer*, é criada uma entrada na tabela de páginas para  $P_i$  e o valor de  $R_i$  é colocado para 0. Sempre que  $P_i$  é referenciada e já está carregada na área de *buffer*, o valor de seu bit  $R_i$  é alterado para 1. Quando uma página na área de *buffer* deve ser substituída, o algoritmo SLRU ler todos os  $R_i$  em uma ordem fixa (menor endereço para o maior endereço). A primeira página encontrada com bit de referência 0 é substituída. Se todos os bits de referências têm valor 1, a página substituída é a que apresenta menor endereço de memória. Considere um mecanismo de gerenciamento de *buffer* com dois níveis, onde a capacidade da região de *buffer* do SGBD é de 3 páginas. A execução de consultas de um grupo de usuários em um sistema de banco de dados referencia sempre cinco páginas distintas  $P_i$ , onde  $i \in \{1, 2, 3, 4, 5\}$ . As páginas estão armazenadas inicialmente em memória secundária (onde estão armazenados os objetos do banco de dados). O fluxo de páginas decorrente da execução dessas consultas é:



Observando o fluxo acima, podemos ver que a primeira página a ser referenciada foi  $P_3$ , a segunda  $P_1$  e assim por diante. **Monte os mapas de alocação da região de *buffer* utilizando as políticas de alocação SLRU e LRU.** Por exemplo, nos instantes  $t_1$  e  $t_2$  a área de *buffer* apresentará o seguinte mapa de alocação:



**Calcule ainda as taxas de acerto para as duas políticas e indique a mais eficiente.**

6. Uma empresa decidiu por utilizar uma estrutura de indexação de arquivo grade (*grid file*) para criar um índice secundário sobre os atributos lotação e salário da tabela Empregado. Os projetistas do banco de dados resolveram particionar o espaço (dimensão) salário nos intervalos: (0,800], (800,1500], (1500,1900], (1900,2700], (2700,3800] e (3800,17000]. Por sua vez, o espaço lotação foi particionado nos intervalos: (0,2], (2,5], (5,10], (10,12] e (12,13]. Com base neste



particionamento, (i) construa a estrutura de diretório, e (ii) calcule quantos acessos a discos são necessários para executar a seguinte consulta:

*Select nome from Empregado  
where salário>2800 and salário<1700 and  
lotação>4 and lotação<12*

7. Determine a altura de uma **árvore  $B^+$**  utilizada para indexar uma tabela de cardinalidade igual a **c** através de índice primário esparsos. As páginas têm capacidade de **x** kbytes. Considere que a chave de busca tem tamanho de **k** bytes. Cada ponteiro necessita de **p** bytes para ser armazenado. Uma tupla da tabela a ser indexada tem tamanho de **t** bytes. Considere o melhor caso.

8. Dado que a operação de seleção é distributiva com relação à operação diferença na álgebra relacional, mostre que a seguinte regra de transformação é verdadeira:

$$\sigma_{\theta}(R \cap S) = \sigma_{\theta}(R) - (\sigma_{\theta}(R) - \sigma_{\theta}(S))$$

9. (a) Quantos acessos a discos são necessários para executar uma operação de escrita no RAID nível 1 com **N** discos?

(b) Quantos acessos a discos são necessários para executar uma operação de escrita no RAID nível 5 com **N** discos de dados?

10. Considere um mecanismo de buffer com dois níveis de memória. A memória  $M_1$  apresenta tempo de acesso igual a  $10^{-7}$  s e taxa de acerto igual a 70%. A memória  $M_2$  está montada através de uma unidade de disco com as seguintes características:

Palavra de 64 bits

Velocidade de rotação do disco: 2400 r/min

Capacidade de armazenamento de cada trilha: 64000 bits

Capacidade de cada setor de uma trilha: 64 bits

Calcule o tempo necessário para o mecanismo de buffer acessar uma palavra. Considere o tempo de procura igual a  $10^{-3}$  s. Suponha que o sistema apresenta um barramento com velocidade de transmissão extremamente alta. Portanto, o tempo gasto para transferir dados através do barramento da unidade de disco até a memória principal pode ser desprezível.

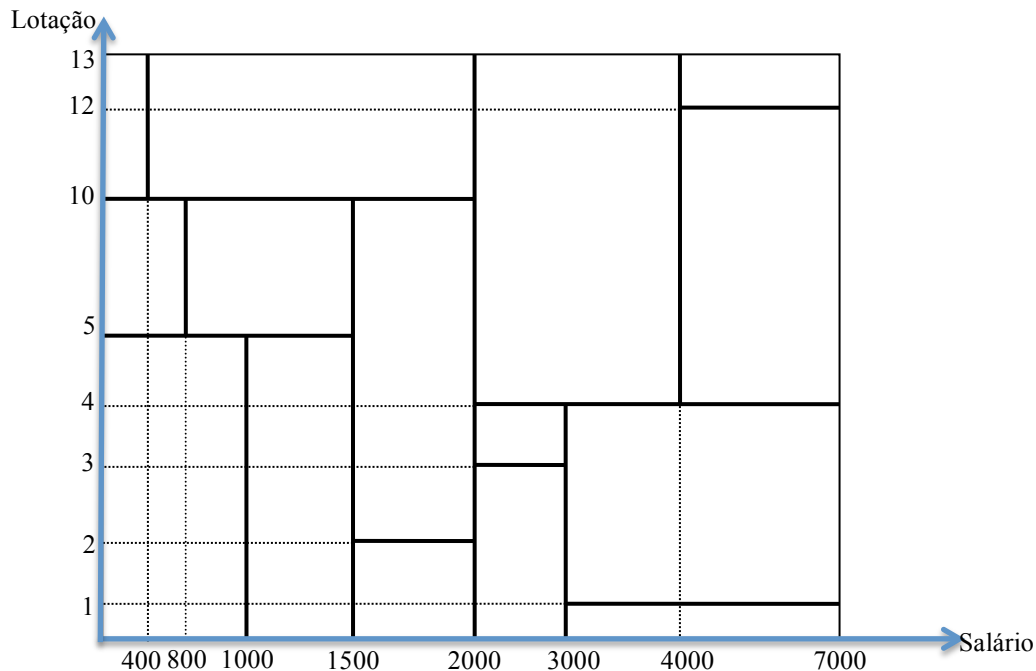
11. Uma empresa decidiu por utilizar uma estrutura de indexação multi-chave para criar um índice secundário sobre os atributos lotação e salário da tabela Empregado. Para tanto, foi decidido utilizar-se uma kd-tree. Os projetistas do banco de dados resolveram particionar o espaço salário x lotação conforme mostrado abaixo. Com base neste particionamento, calcule quantos acessos a discos são necessários para executar as seguintes consultas:

(a) *Select nome from Empregado*

*where salário>500 and salário<2000 and lotação>4 and lotação<11*



(b) Select nome,salario from Empregado  
where lotação>7



12. Uma empresa decidiu por utilizar uma estrutura de indexação multi-chave para criar um índice secundário sobre os atributos lotação e salário da tabela Empregado. Para tanto, foi decidido utilizar-se uma **quad-tree**. O espaço bi-dimensional *salário X lotação* foi particionado em sub-espacos, conforme mostrado na figura abaixo. Calcule quantos acessos a discos são necessários para executar as seguintes consultas:

(a) Select nome from Empregado

where salário>2000 and salário<=4000 and lotação>5 and lotação<16

(b) Select nome,salario from Empregado where lotação>10

