



## Exercícios de Estruturas de Dados II

Prof. Saulo Queiroz

Nome e R.A. do Aluno(a): \_\_\_\_\_

Data da prova: \_\_\_\_/\_\_\_\_/\_\_\_\_

**OBS.: Respostas devem ser dadas na mesma folha da questão. Apresente as suposições se necessário.**

Assinatura<sup>1</sup> do Aluno(a): \_\_\_\_\_

1. Ilustre a árvore AVL resultante da sequência de inserção 3, 1, 2, 7, 10, 11, 12.
2. A função de espalhamento  $f(x) = x \% M$  para uma tabela com  $M$  posições é conhecida como método da divisão.
  - (a) Qual a fórmula da função hash conhecida como método da multiplicação?
  - (b) Qual a recomendação de Donald Knuth para configuração do método da multiplicação?
3. Cite duas técnicas clássicas para tratamento efetivo de colisão em tabelas hash.
4. Pesquise e reporte o significado de fator de carga em tabelas de espalhamento. Como o fator de carga explica o desempenho de tempo de uma tabela hash com sondagem linear? E o de espaço?
5. O fator de carga no hash com endereçamento aberto é no máximo 1. É possível termos um fator de carga maior do que 1 no hash com encadeamento (isto é, que usa listas encadeadas para tratar colisão)? Por que? Justifique apresentando ao menos um exemplo.
6. Considere a seguinte afirmação: “o objetivo do hash com encadeamento é remover a necessidade de estimarmos antecipadamente a quantidade máxima de chaves que se pretende armazenar pois ele aloca espaço dinamicamente para cada chave”. Essa afirmação está correta? Critique.
7. Qual a recomendação do prof. Robert Sedgewick [1] para o fator de carga de tabelas hash com encadeamento? Como interpretar tal recomendação para o desempenho de tempo e espaço do hash?. [1] Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching
8. Implemente em C os algoritmos abaixo descritos. Defina você mesmo os parâmetros necessários à sua função.
  - (a) inserção de chave inteira hash com encadeamento
  - (b) hash com endereçamento aberto e sondagem linear com método da divisão
  - (c) hash com endereçamento aberto e sondagem quadrática com método da multiplicação
9. Você foi solicitado avaliar o desempenho de busca de uma dada função de espalhamento (*hash*) com sondagem linear (*linear probing*) de 1 milhão de chaves de 32 bits. Para cada busca realizada, você registrou a quantidade de colisões experimentadas e plotou a distribuição de frequência correspondente a essa quantidade (Fig 1). Com base nesse resultado responda as questões:

---

<sup>1</sup> atesta estar de acordo com correção. Assinar após divulgação da nota ou no dia da realização da prova caso opte pelo uso de lápis.

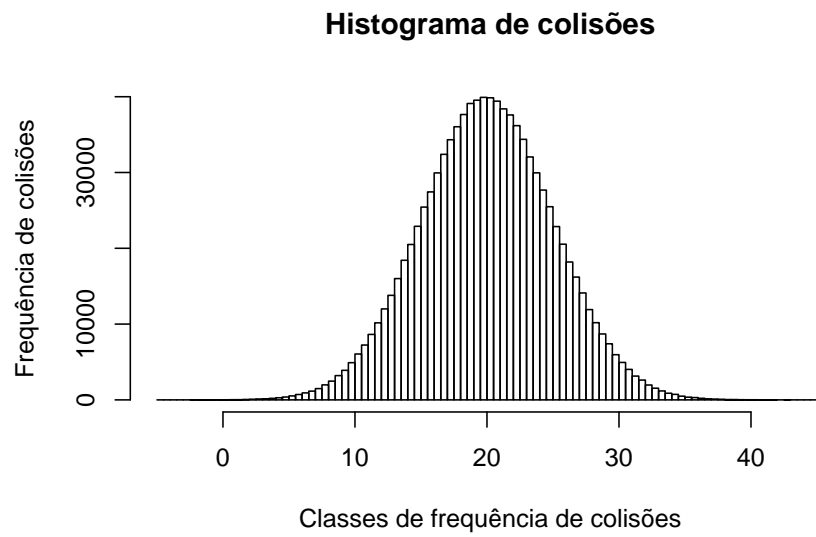


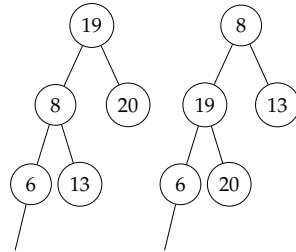
Figura 1: Histograma das quantidade de colisões num *hash* com sondagem linear.

(a) O *hash* pode ser considerado eficiente? Por que? Argumente!

- (b) Assumindo que o *hash* mantém o mesmo desempenho médio para uma quantidade arbitrária  $n$  de nós, e tendo em conta o número de comparações realizadas numa busca, sob que condições o *hash* avaliado pode ser considerado melhor que uma árvore AVL? Assuma que a altura da AVL é  $1.44 \log_2 n$ .

10. Um colega seu implementou uma função de inserção em uma árvore AVL. A função só não ficou impecável por um único detalhe: nas ocasiões em que era necessário invocar o procedimento de rotação simples à direita ela invocava o procedimento de rotação simples à esquerda. A Figura 10 dá um mero exemplo do impacto desse equívoco. A árvore  $T_1$  tornou-se em  $T_2$  após a incorreta rotação. Sua missão é fazer um procedimento que receba como entrada a árvore incorreta (por exemplo,  $T_2$ ) e corrija-a para o formato que ela deveria ter se a rotação tivesse sido adequadamente aplicada. Assuma que a entrada é o ponteiro do ponteiro para o nó pai do nó incorretamente rotacionado (no exemplo esse nó pai é o 8 na árvore  $T_2$ ). Assuma que seu colega não errou na atualização do campo de fator de balanceamento.

Figura 2: (Sub-)Árvores AVL  $T_1$  (inicial) e  $T_2$  (errada), respectivamente.



11. Faça um procedimento que recebe três parâmetros de entrada: uma Árvore Binária de Pesquisa (ABP), uma árvore AVL e um valor inteiro  $c$  representando uma constante maior que zero. Se a altura  $h$  da ABP for logarítmica, imprima na tela A altura  $h$  da ABP é  $O(\log n)$ . Assuma que a quantidade  $n$  de nós de cada árvore é igual e que uma ABP tem altura logarítmica se  $h \leq c \cdot \log n$ . DICA: use duas funções. DICA 2: utilize uma variável para contar níveis e outra para registrar a maior altura conhecida.

12. Para uma determinada aplicação foi possível prever como cresceriam as respectivas alturas de uma árvore AVL e de uma ABP sob uma determinada quantidade  $n$  de chaves. Essas alturas são ilustradas na Figura 3. Nesse contexto, informe a(s) circunstância(s) em que a ABP seria preferível à AVL para obtermos um melhor tempo de execução mesmo levando em conta o resultado do gráfico. Justifique sua resposta.

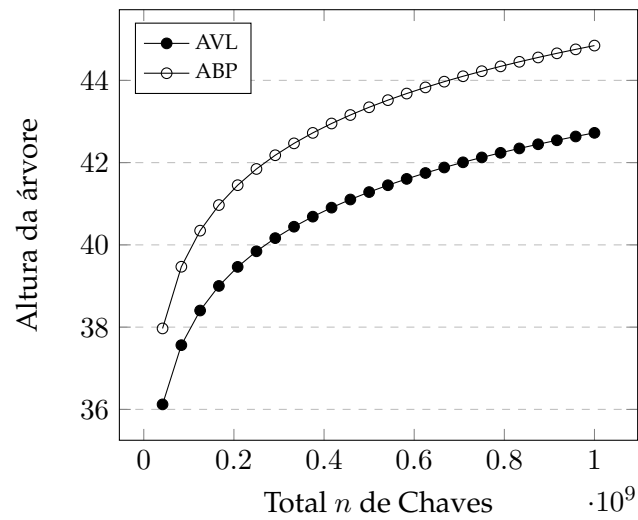


Figura 3: AVL vs. ABP: Comparação de altura sob  $n$ .

EXTRA Pesquise e explique a ideia geral de ao menos um dentre as seguintes soluções hash: Hopscotch hashing, Robin Hood hashing, last-come-first-served hashing and cuckoo hashing.

EXTRA

13. Pesquise e apresente um algoritmo para transformar uma string em um número inteiro único que pode ser utilizado como entrada de uma função de espalhamento. Explique a ideia do algoritmo e, se necessário, apresente as suposições do código.