

Lista de Exercícios – Módulos 3 e 4

Introdução à Ciência de Computação II

Prof. Moacir Ponti

8 de dezembro de 2020

1. Codifique na linguagem C uma função que receba uma matriz com $n \times m$ elementos do tipo unsigned char, e que altere essa matriz de forma a alterar o intervalo dinâmico dos valores da matriz. Seja $[a, b]$ os valores mínimo (a) e máximo (b) na matriz, a função deverá processar cada elemento de forma que a matriz agora tenha valores no novo intervalo $[c, d]$. Para isso, cada valor r da matriz deve ser processado utilizando a transformação abaixo:

$$T(r) = (r - a) \left(\frac{d - c}{b - a} \right) + c$$

Considerando as operações aritméticas e acessos à matriz como relevantes, faça a análise de complexidade da função.

2. Analise o algoritmo abaixo em termos da complexidade de tempo e memória utilizada

```
int* primos(int n) {
    int* p = calloc(n+1, sizeof(int));
    p[1] = 1;
    for (i = 2; i <= n; i++) {
        for (j = i-1; j > 1; j--) {
            if (i % j == 0) break;
        }
        if (j == 1) p[i] = 1;
    }
    return p;
}
```

3. Compare dois algoritmos que possuem as seguintes funções de eficiência, em termos de operações relevantes representadas pelas constantes $a, b \in \mathbb{Z}^*$, sendo que sabemos que $b > a$. As funções são dadas por:

$$f(n) = 5bn + 10bn$$
$$g(n) = a[n \log_2(n)] + b$$

Compare as funções do ponto de vista de suas funções completas considerando as constantes, e também de sua eficiência assintótica.

4. No caso de ordenar um vetor de números `double` utilizando Radixsort, considere que particionamos o `double` em sua parte inteira (antes da vírgula) e fracionária (depois da vírgula). Então, interpretamos o valor antes da vírgula como `long int` e particionamos em 4 blocos e depois interpretamos os valores depois da vírgula como `int` e particionamos também em 4 blocos.

Faça a análise assintótica considerando explicitamente a configuração das partições, considerando que utilizamos o counting sort como subrotina.

5. Ao projetar uma função para busca binária por interpolação, qual seria a principal vantagem e o principal risco?
6. Qual o principal objetivo de se projetar um método Hashing para um determinado problema?
7. Considere uma tabela hash com endereçamento aberto de tamanho $m = 13$.
- Realize a inserção dos elementos 105, 13, 26, 11, 49, 301, 6, 1, utilizando:
 - O método da divisão e tratamento de colisão com sondagem linear $h(k) = (k + i) \bmod m$
 - O método da divisão e tratamento de colisão com sondagem quadrática $h(k) = (k + i^2) \bmod m$
 - O método da multiplicação com $A = 0,15$ e tratamento de colisão com sondagem quadrática
 - Qual o fator de carga da tabela da questão anterior, após as inserções? Quantas operações são esperadas para essa tabela, ao realizar uma busca por um elemento?
8. Repita o exercício anterior, mas agora utilizando uma tabela hash de tamanho $m = 5$ com tratamento de colisão por encadeamento externo. Responda:
- Qual o fator de carga final da tabela?
 - Para essa tabela especificamente, quantas comparações serão feitas no pior caso para retornar uma busca sem sucesso?
9. A definição do tamanho m da tabela hash é parte importante no projeto de tabelas hash para evitar colisões e diminuir o tempo de execução para inserções e buscas. Sugira valores de m para o número de elementos a serem inseridos n abaixo, sendo que n é o número esperado de elementos a serem armazenados na tabela:
- $n \sim 1000$ e endereçamento aberto
 - $1.000 \leq n \leq 100.000$ e endereçamento aberto
 - $n \sim 1.000.000$ e endereçamento aberto
 - $100 \leq n \leq 1.000$ e encadeamento externo
 - $n \sim 1.000.000$ e encadeamento externo
10. Qual a vantagem do uso de Hashing universal na montagem de uma tabela hash?