

**Universidade Tecnológica Federal do Paraná (UTFPR)**  
**Departamento Acadêmico de Informática (DAINF)**  
**Professora: Juliana de Santi**

## Lista de exercícios (Hash)

1) (Cormem) Faça no papel a inserção das chaves 5, 28, 19, 15, 20, 33, 12, 17, 10 em uma tabela hash com colisões resolvidas por encadeamento. Seja a tabela com 9 posições, e seja a função hash  $h(k) = k \bmod M$ .

2) (Cormem) Faça no papel a inserção das chaves 10, 22, 31, 4, 15, 28, 17, 88, 59 em uma tabela hash de comprimento  $m = 11$  usando o endereçamento aberto com a função hash primário  $h'(k) = k \bmod m$ . Ilustre o resultado destas inserções com a sondagem linear, com a sondagem quadrática com  $c_1 = 1$  e  $c_2 = 3$ , e com a utilização do hash duplo com  $h_2(k) = 1 + (k \bmod (m - 1))$ . Indique o número total de colisões para cada técnica.

3) Vimos em aula, que todos os exemplos usaram chaves numéricas. O que fazer se a chave é uma cadeia de caracteres? Como calcular uma função de espalhamento nesse caso? Na realidade é muito fácil. Cada caractere é um número entre 0 e 255. Portanto, uma cadeia não vazia pode ser interpretada como a representação em base 256 de um número. Suponha que  $s$  é uma cadeia de comprimento 2. Então o número correspondente é

$$s[0] * 256 + s[1]$$

Por exemplo, se  $s$  é “AB” então o número correspondente é  $65 * 256 + 66$ , ou seja, 16706. Para fazer os cálculos de maneira eficiente, basta usar o método de Horner:

```
int hash_function (char *v, int M) {  
    int i, h = v[0];  
    for (i = 1; v[i] != '\0'; i++)  
        h = (h * 256 + v[i]) % M;  
    return h;  
}
```

Faça um programa para inserir os nomes abaixo em uma tabela hash com colisões resolvidas por encadeamento, para uma tabela com 7 posições ( $M = 7$ ). Utilize como modelo o código **hash\_string.c** em anexo.

Marina  
Pedro  
Joana  
Thais  
Fabio  
Jonas  
Joaquim  
Mauricio  
Jorge  
Ana  
Patricia  
Henrique