

Pesquisa em memória primária: hashing

Algoritmos e Estruturas de Dados II

Hashing Dinâmico

- ▶ Evita que o fator de carga fique muito alto (N/M)
- ▶ **Ideia:** dobra o tamanho da tabela hashing quando fator de carga passa de um limiar.

```
void expande(TipoPesos p, TipoDicionario T) {  
  
    T2 = NovoHashing(2 * M);  
    for (i = 0; i < M; i++)  
        if (((strcmp(T[i].Chave, Vazio) != 0) &&  
            (strcmp(T[i].Chave, Retirado) != 0))  
            InsererHashing(T[i].Chave, p, T2);  
}  
  
/* libera a memória para T */
```

$\alpha = N / M$	C(n)
0.1000	1.0556
0.2000	1.1250
0.3000	1.2143
0.4000	1.3333
0.5000	1.5000
0.6000	1.7500
0.7000	2.1667
0.8000	3.0000
0.9000	5.5000
0.9500	10.5000
0.9800	25.5000
0.9900	50.5000

Hashing Duplo

- ▶ Mecanismo para resolução de colisão
- ▶ Endereçamento aberto apresenta o efeito de agrupamento
 - ▶ Agrupa chaves próximas a um endereço, mesmo que a função de espalhamento não a coloque lá.
- ▶ Solução: hashing duplo
 - ▶ Ao invés de examinar cada uma das posições sucessivas após uma colisão, uma segunda função de hashing é aplicada.

Hashing Duplo

```
Apontador Pesquisa(TipoChave Ch, TipoPesos p, TipoDicionario T) {
    unsigned int  i=0, count = 0;
    unsigned int  Inicial, segundo;

    Inicial = h(Ch, p);      /* transforma a chave */
    segundo = h2(Inicial);   /* h2(x) = ((x%97)+1) */

    while ((strcmp(T[(Inicial + i) % M].Chave, Vazio) != 0) &&
           (strcmp(T[(Inicial + i) % M].Chave, Ch) != 0) &&
           (count < M)) {
        i += segundo; count++;
    }

    if (strcmp(T[(Inicial + i) % M].Chave, Ch) == 0)
        return ((Inicial + i) % M);
    else
        return M;    /* Pesquisa sem sucesso */
}
```

Hashing Duplo: Análise

$$C(n) = \frac{1}{\alpha} \ln \left(\frac{1}{1 - \alpha} \right)$$

$\alpha = N / M$	$C(n)$
0.1000	1.0536
0.2000	1.1157
0.3000	1.1889
0.4000	1.2771
0.5000	1.3863
0.6000	1.5272
0.7000	1.7200
0.8000	2.0118
0.9000	2.5584
0.9500	3.1534
0.9800	3.9919
0.9900	4.6517

Hashing Duplo: Análise

Hashing duplo:

$$C(n) = \frac{1}{\alpha} \ln \left(\frac{1}{1-\alpha} \right)$$

Endereçamento aberto:

$$C(n) = \frac{1}{2} \left(1 + \frac{1}{1-\alpha} \right)$$

	Hashing duplo	Endereçamento aberto linear
$\alpha = N / M$	$C(n)$	$C(n)$
0.1000	1.0536	1.0556
0.2000	1.1157	1.1250
0.3000	1.1889	1.2143
0.4000	1.2771	1.3333
0.5000	1.3863	1.5000
0.6000	1.5272	1.7500
0.7000	1.7200	2.1667
0.8000	2.0118	3.0000
0.9000	2.5584	5.5000
0.9500	3.1534	10.5000
0.9800	3.9919	25.5000
0.9900	4.6517	50.5000