

## Resolução do Exercício

Utilizando a função hash:

$$h(\text{key}) = \text{key} \bmod 11$$

Armazene os seguintes valores em sequência: 82,31,28,4,45,27,59,79,35  
na tabela hash :

```
int table[11];
```

Realize as mesmas operações variando os seguintes métodos:

- linear rehashing
- quadratic rehashing
- external chaining
- coalesced chaining (cellar size=4 e hash function  $h(k) = k \bmod 7$ )

Para cada uma das estratégias de resolução de colisão determine:

- load factor
- média do número de testes (probes) para encontrar valor presente na tabela

---

Calculando a função  $h(k) = k \% 11$  para todos os valores da sequência temos :

key	82	31	28	4	45	27	59	79	35
h(key)	5	9	6	4	1	5	4	2	2

Observando a tabela acima haverá colisão apenas a partir da chave 27, que colidirá com a chave 82 na posição 5 do vetor.

### a. Utilizando linear probing para resolução de colisão temos:

$h'(k,i) = (k + a*i) \bmod 11$ , onde escolhemos  $a=1$  e  $i \geq 1$  variando iterativamente a cada tentativa de se resolver a colisão.

$h(27)=5$  colisão  $h(82)$

$h'(27,1)=6$  colisão  $h(28)$

$h'(27,2)=7$  ok

$h(59)=4$  colisão  $h(4)$

$h'(59,1)=5$  colisão  $h(82)$

$h'(59,2)=6$  colisão  $h(28)$

$h'(59,3)=7$  colisão  $h'(27)$

$h'(59,4)=8$  ok

$h(79)=2$  ok

$h(35)=2$  colisão  $h(79)$

$h'(35,1)=3$  ok

### b. Utilizando quadratic probing para resolução de colisão temos:

$h'(k) = (k + a*i^2) \bmod 11$ , onde escolhemos  $a=1$  e  $i \geq 2$  (pois  $i=1$  recai em linear probing)

$h(27)=5$  colisão  $h(82)$

$h'(27,2)=9$  colisão  $h(31)$

$h'(27,3)=3$  ok

$h(59)=4$  colisão  $h(4)$

$h'(59,2)=8$  ok

$h(79)=2$  ok

$h(35)=2$  colisão  $h(79)$

$h'(35,2)=6$  colisão  $h(28)$

$h'(35,3)=0$  ok

### c. Utilizando external chaining para resolução de colisão temos:

Cria-se uma lista ligada em cada posição do tabela hash.

```
idx[0] = NULL
idx[1] = 45 -> NULL
idx[2] = 79 -> 35 -> NULL
idx[3] = NULL
idx[4] = 4 -> 59 -> NULL
idx[5] = 82 -> 27 -> NULL
idx[6] = 28 -> NULL
idx[7] = NULL
idx[8] = NULL
idx[9] = 31 -> NULL
idx[10] = NULL
```

### d. Utilizando coalesced chaining para resolução de colisão temos:

Neste caso modificaremos a função básica de hash para  $h(k) = k \bmod 7$ , cujo efeito colateral será manter as 4 últimas posições ( $4=11-7$ ) do vetor vazias para serem reaproveitadas na acomodação de colisões.

Também será necessário alocar um segundo vetor cuja finalidade é indicar qual o próximo índice do vetor original contém a resolução da colisão.

$idx[]$  é a tabela hash  $\therefore next[]$  indica próximo índice em  $idx[]$

Obs: Todo o vetor  $next[]$  é inicializado com valor especial=VAZIO!

$h(82)=5 \therefore h(31)=3 \therefore h(28)=0 \therefore h(4)=4$

$h(45)=3$  colisão com  $h(31)=3$

como  $next[3]$  está vazio

escolhemos a última posição livre  $idx[10]=45$

e apontamos  $next[3]=10$

$h(27)=6$  ok

$h(59)=3$  colisão com  $h(31)=3$

como  $next[3]=10$  utilizado para  $h(45)$

escolhemos a posição livre  $idx[9]=59$

e apontamos  $next[10]=9$

Assim a sequência de resolução final é

$h(59)=3 \rightarrow next[3]=10 \rightarrow next[10]=9$

$h(35)=0$  colisão com  $h(28)$

como  $next[0]$  está vazio

escolhemos a última posição livre  $idx[8]=35$

e apontamos  $next[0]=8$

**e. Load Factor = numero de elementos na tabela / tamanho da tabela**

Pela definição o load factor não depende do método. Após a inserção da sequência de 9 elementos numa tabela de tamanho 11, o load factor será  $9/11=0.81$ .

Isto significa que a tabela está 80% preenchida. Exceção feita para o caso de external chaining onde a definição de load factor não faz sentido, uma vez que o espaço de armazenamento cresce dinamicamente.

**f. média do número de testes (probes) para encontrar valor presente na tabela**

Basta somar todas as linhas coloridas do texto acima para saber o total de consultas.

Note que os valores que não sofreram colisão devem somar como 1 consulta para cada.

linear probing: 16 consultas / 9 valores = 1.777...

quadratic probing: 14 consultas / 9 valores = 1.555...

external chaining: 12 consultas / 9 valores = 1.333...

coalesced chaining: 12 consultas / 9 valores = 1.333...