

## Trabalho Estrutura de Dados

Disciplina: Estruturas de Dados

Professor:: Welington Gadelha

Acadêmico: Inácio Radin Rigatti

Código disponível em: <https://github.com/inaciorigatti/HashMap>

- 1) O mapa Hash que conseguimos quando M possui o valor de 10:

```
PROBLEMS 1 OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS C:\Users\inaci\OneDrive\Desktop\hashmap> & 'c:\Users\inaci\OneDrive\Desktop\hashmap\Microsoft-MIEngine-Out-ipbvi4px.vfi' '--stderr=Microsoft-MIEngine-Out-m0xyuwak.juc'
[0]: [10]
[1]: [31] -> [21] -> NULL
[2]: NULL
[3]: [3]
[4]: [14]
[5]: NULL
[6]: [26]
[7]: [17]
[8]: [8]
[9]: [29]
PS C:\Users\inaci\OneDrive\Desktop\hashmap>
```

Já quando M possui o valor 13:

```
PS C:\Users\inaci\OneDrive\Desktop\hashmap> & 'c:\Users\inaci\OneDrive\Desktop\hashmap\Microsoft-MIEngine-Out-m0xyuwak.juc' '--stderr=Microsoft-MIEngine-Out-m0xyuwak.juc'
[0]: [26]
[1]: [14]
[2]: NULL
[3]: [29] -> [3] -> NULL
[4]: [17]
[5]: [31]
[6]: NULL
[7]: NULL
[8]: [8] -> [21] -> NULL
[9]: NULL
[10]: [10]
[11]: NULL
[12]: NULL
PS C:\Users\inaci\OneDrive\Desktop\hashmap>
```

Observamos que mesmo que o número de colisões tenha sido ligeiramente maior com  $M = 13$ , a distribuição das chaves foi mais uniforme, com mais espaços livres e melhor separação entre os elementos. Quando a tabela tem um tamanho primo,

evita-se padrões cíclicos de agrupamento que ocorrem com tamanhos compostos (como 10), melhorando a performance média da busca e inserção.

- 2) A função de hash proposta distribuiu de forma razoável as chaves, mas concentrou múltiplas colisões em poucos índices.

```
PROBLEMS 2 OUTPUT TERMINAL PORTS DEBUG
PS C:\Users\inaci\OneDrive\Desktop\hashmap> &
osoft-MIEngine-Out-qmsihdpz.hz3' '--stderr=Micro
[0]: [25] -> [14] -> [3] -> NULL
[1]: [9]
[2]: [37]
[3]: NULL
[4]: NULL
[5]: NULL
[6]: [50]
[7]: NULL
[8]: [7] -> [18] -> [62] -> NULL
[9]: NULL
[10]: [30] -> [41] -> NULL
PS C:\Users\inaci\OneDrive\Desktop\hashmap>
```

3)

```
osoft-MIEngine-Out-nmdlaged.uus' '--stderr=Microso
Tabela cheia.
[0]: [i]
[1]: [s]
[2]: [c]
[3]: [i]
[4]: [l]
[5]: [o]
[6]: [r]
[7]: [i]
[8]: [o]
[9]: [i]
[10]: [n]
[11]: [a]
[12]: [t]
[13]: [g]
[14]: [a]
[15]: [m]
PS C:\Users\inaci\OneDrive\Desktop\hashmap> █
```

4) A inserção dos números, posteriormente o hashmap com a remoção de 2 e depois a adição de 17 e 2.

```
PS C:\Users\inaci\OneDrive\Desktop\hashmap> & '
osoft-MIEngine-Out-f5owgck3.d3e' '--stderr=Micro
[0]: [33]
[1]: [0]
[2]: [12]
[3]: [31]
[4]: NULL
[5]: [27]
[6]: [5]
[7]: NULL
[8]: [8]
[9]: [19]
[10]: [21]

[0]: [33]
[1]: [0]
[2]: [12]
[3]: [31]
[4]: NULL
[5]: NULL
[6]: NULL
[7]: NULL
[8]: [8]
[9]: [19]
[10]: [21]

[0]: [33]
[1]: [0]
[2]: [12]
[3]: [31]
[4]: [2]
[5]: NULL
[6]: [17]
[7]: NULL
[8]: [8]
[9]: [19]
[10]: [21]
PS C:\Users\inaci\OneDrive\Desktop\hashmap> █
```

5)

PROBLEMS    OUTPUT    TERMINAL    PORTS    DEBUG CONSOLE

```
'Python' inserida na tabela com sucesso.  
'Java' inserida na tabela com sucesso.  
'C' inserida na tabela com sucesso.  
'Go' inserida na tabela com sucesso.  
'Kotlin' inserida na tabela com sucesso.  
'Swift' inserida na tabela com sucesso.  
'Rust' inserida na tabela com sucesso.  
'Ruby' inserida na tabela com sucesso.  
'PHP' inserida na tabela com sucesso.  
'TypeScript' inserida na tabela com sucesso.
```

=== MENU ===

1. Inserir nova linguagem
2. Remover linguagem
3. Imprimir tabela
0. Sair

Escolha: 3

```
[ 0]: NULL  
[ 1]: NULL  
[ 2]: [Go] -> [Python] -> NULL  
[ 3]: NULL  
[ 4]: NULL  
[ 5]: [Swift] -> [Kotlin] -> NULL  
[ 6]: [Java] -> NULL  
[ 7]: [TypeScript] -> [C] -> NULL  
[ 8]: NULL  
[ 9]: NULL  
[10]: [Rust] -> NULL  
[11]: NULL  
[12]: [PHP] -> NULL  
[13]: NULL  
[14]: NULL  
[15]: NULL  
[16]: NULL  
[17]: NULL  
[18]: [Ruby] -> NULL  
[19]: NULL
```