**SCC0220 - Laboratório Introdução à Ciência da Computação II**

**Relatório - Aula 10**

**Alunos NUSP**

Pedro Henrique de Sousa Prestes 15507819

Pedro Lunkes Villela 15484287

**Trabalho 10 - MSC**

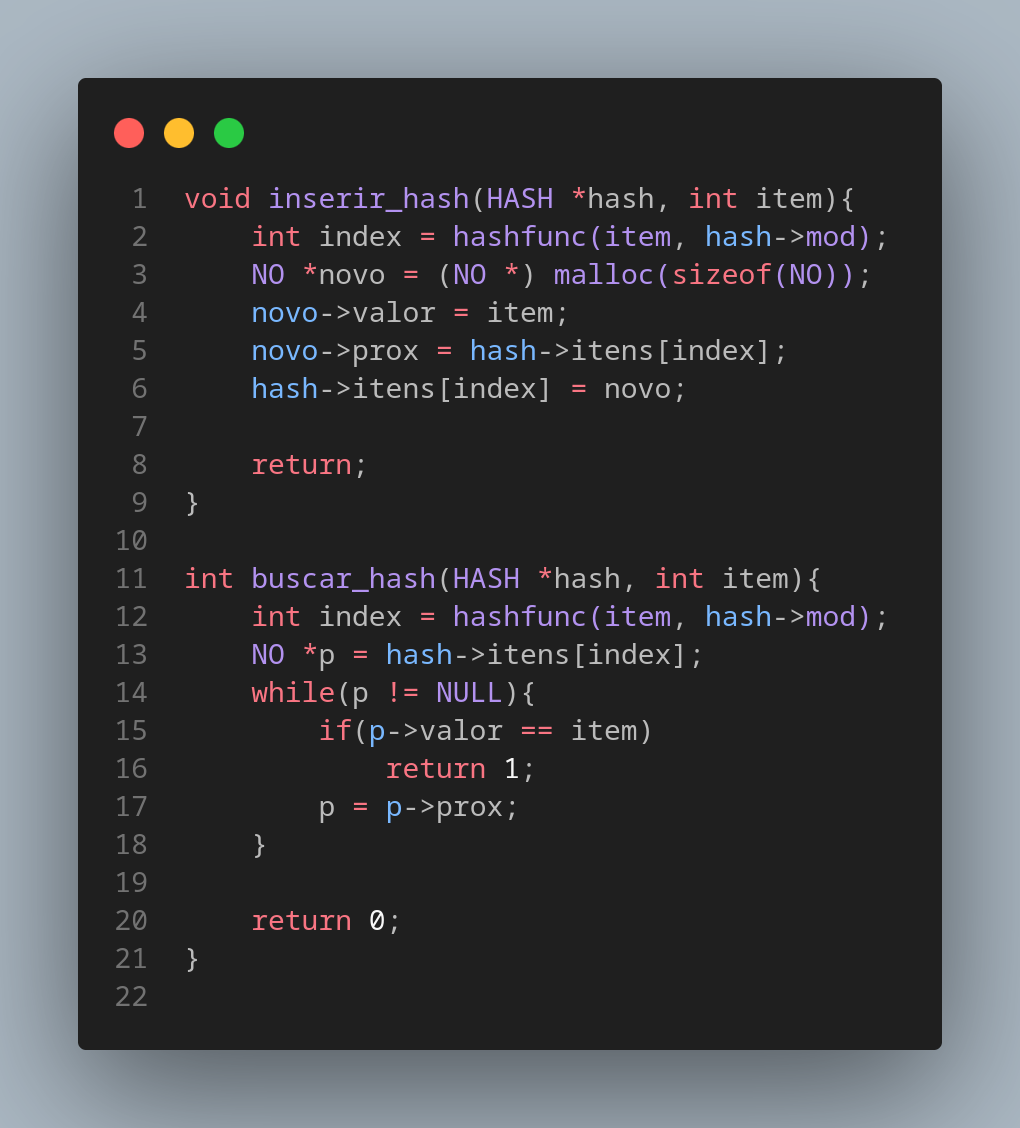
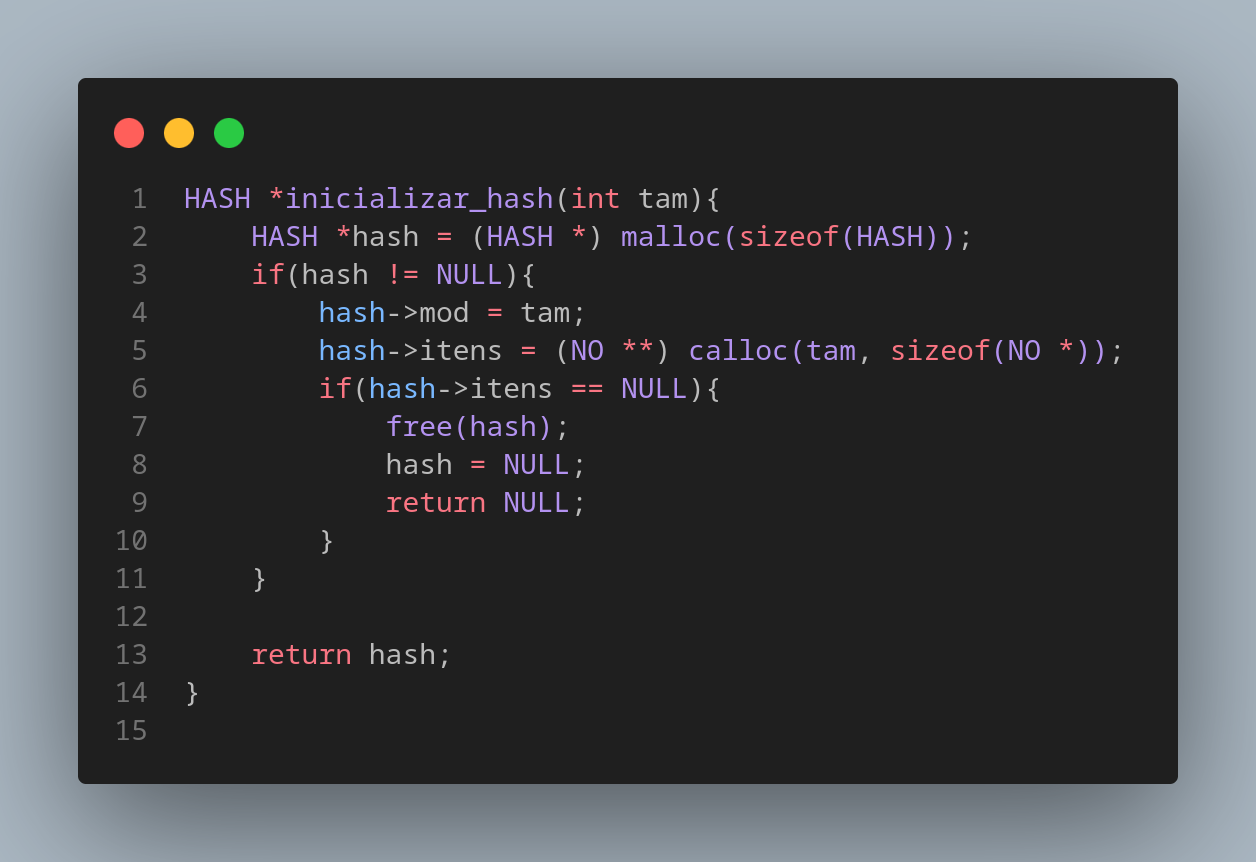
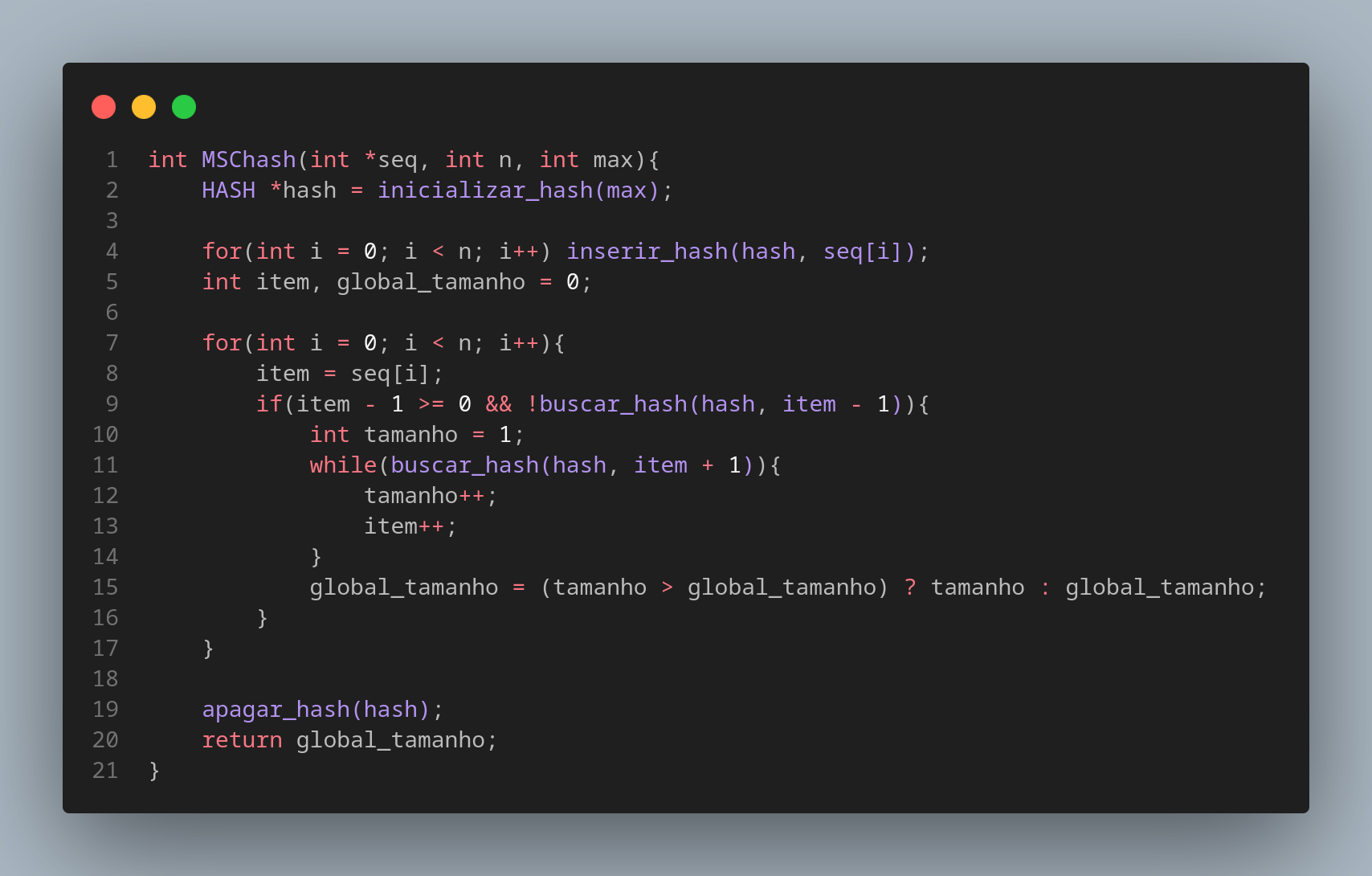
**Hash x Ordenação**

**🡺 Comentário**

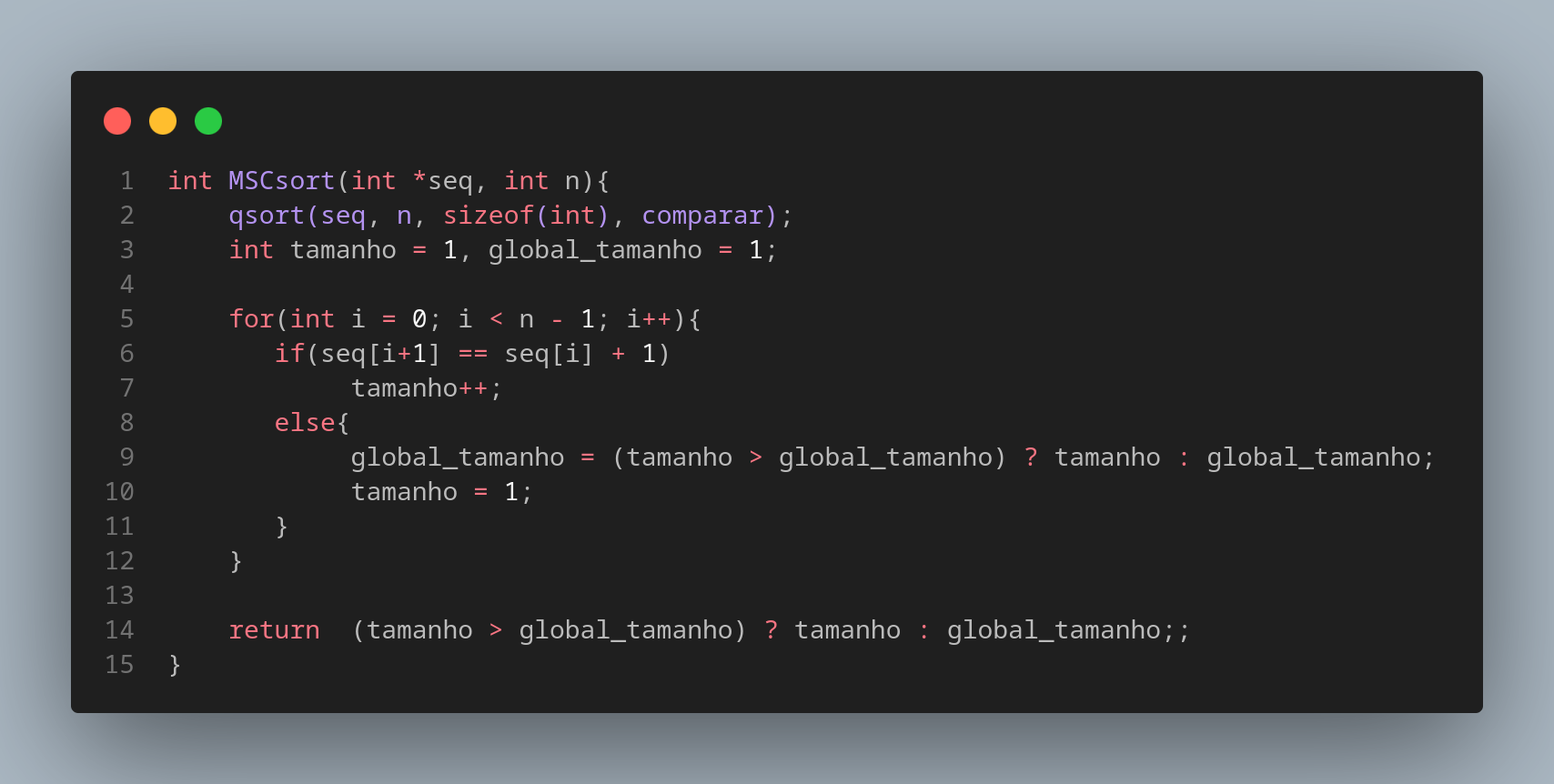
❖ **Hash:** a hash guarda as entradas em um vetor com base no valor de entrada. Com ela também é possível encontrar a posição com uma função hash, o que torna os acessos aos dados O(1). Por esse motivo, na resolução do problema, o programa procura por chaves não salvas na hash, e determina intervalos crescentes a partir delas. No problema, sua complexidade é O(n)

❖ **Ordenação:** o algoritmo linear ordena o vetor e o percorre procurando por sequências crescentes, e anotando seus tamanhos, para no fim retornar o maior. Sua complexidade é O(nlog(n)), já que estamos usando a função qsort() para fazer a ordenação.

**🡺 Código**

****

(Códigos para a hash)

****

(Código para o algoritmo linear)

**🡺 Saída**

Seguem alguns casos testados comparando o desempenho de ambos algoritmos.

**1° Caso**

Tempo de execução Hash: 0.0017s

Tempo de execução Ordenação: 0.0016s

**2° Caso**

Tempo de execução Hash: 0.0021s

Tempo de execução Ordenação: 0.0016s

**3° Caso**

Tempo de execução Hash: 0.0019s

Tempo de execução Ordenação: 0.0016s

**4° Caso**

Tempo de execução Hash: 0.0020s

Tempo de execução Ordenação: 0.0016s

**5° Caso**

Tempo de execução Hash: 0.0026s

Tempo de execução Ordenação: 0.0017s

**6° Caso**

Tempo de execução Hash: 0.0100s

Tempo de execução Ordenação: 0.0035s

**7° Caso**

Tempo de execução Hash: 0.0468s

Tempo de execução Ordenação: 0.0125s

**8° Caso**

Tempo de execução Hash: 0.1136s

Tempo de execução Ordenação: 0.0287s

**9° Caso**

Tempo de execução Hash: 0.7326s

Tempo de execução Ordenação: 0.1502s

**🡺 Conclusão**

O hash, apesar de teoricamente ser mais rápido, quando não há conflito na tabela - O(n) - é mais lento por conta da inicialização da tabela hash e da grande quantidade de alocações necessárias. Em contrapartida, o algoritmo linear será sempre O(nlog(n)). Por essa razão seus gráficos são parecidos, mesmo com a pequena diferença de tempo enre eles.

