



Universidade Federal do Piauí

Alunos: Jhoisnáyra Vitória Rodrigues de Almeida

Pedro Gonçalves Neto.

Curso: Ciência da Computação.

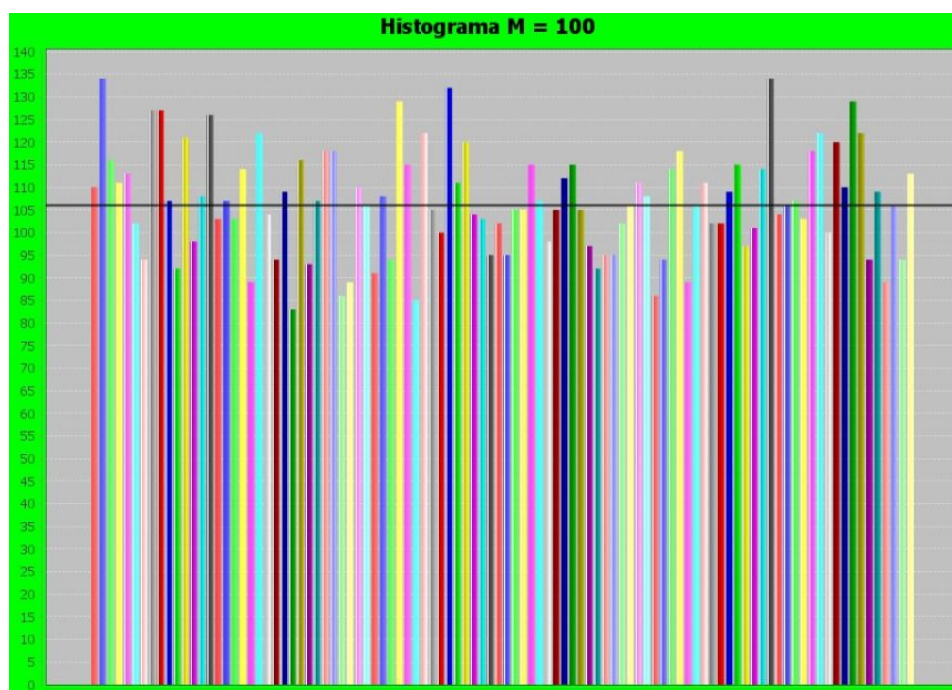
Relatório de Estrutura de Dados:  
Tabela Hash.

Teresina, 14 de outubro de 2019.

- 1) **Eliminar as palavras repetidas do arquivo tale.txt. Depois, fazer um histograma como o mostrado em sala, mas usando  $M = 100$ .**

Na implementação do programa, o arquivo foi lido pela cmd e guardado em um vetor `a[]`. Para não permitir a repetição de palavras, atribui-se os valores do vetor a um HashSet, cuja uma das características é a não admissão de elementos repetidos.

Após isso, usa-se a função de hash ensinada em aula e disponível no livro Java Algorithms para retornar a chave de cada elemento presente no HashSet e adicionar a um vetor novo. Posteriormente, instancia-se um gráfico com JFreeChart passando como parâmetro o vetor com as chaves e o limite (linha amarela no gráfico da imagem 1).



**Imagem 1 - Histograma da Tabela hash em que  $M = 100$ .**

- 2) **Fazer um histograma para as palavras do arquivo tale.txt (sem repetições), com  $M = 97$ , mas com alguma função de hashing diferente da usada na questão anterior.**

Usando uma função de hashing diferente, também apresentada em aula, e com  $M = 97$ , obtêm-se o histograma abaixo.

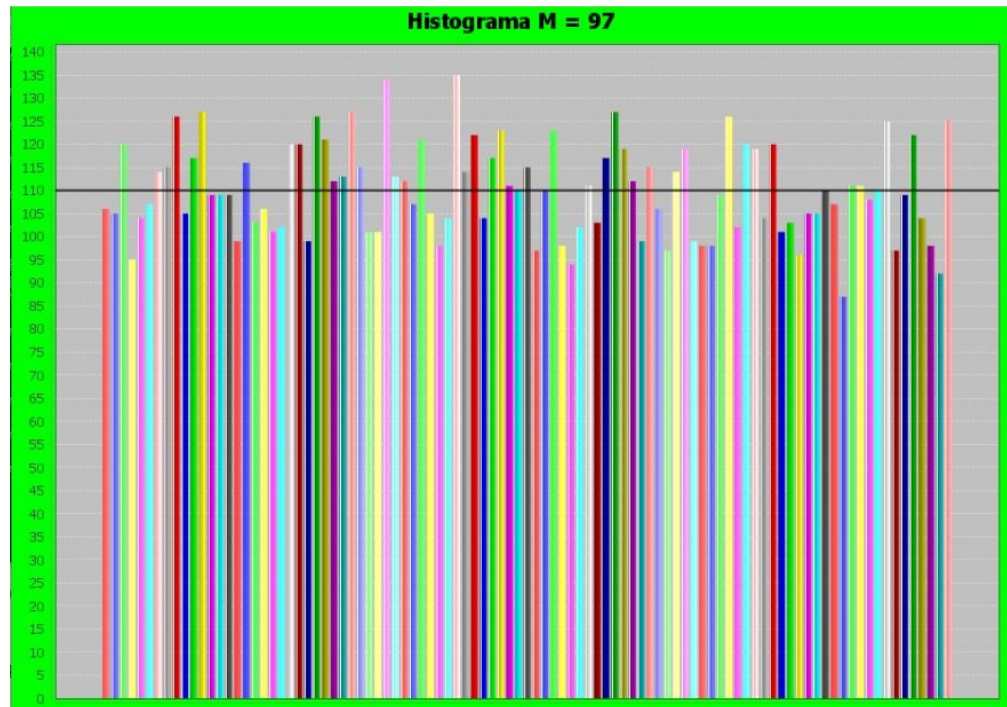


Imagem 2 - Histograma da Tabela hash em que  $M = 97$ .

- 3) Repetir os experimentos com o livro tale.txt usando valores de  $M$  diferentes de 97 (tentar valores que são potência de 2, por exemplo).

Usando potências de 2, aplicou-se  $2^7$  no gráfico da imagem 3:

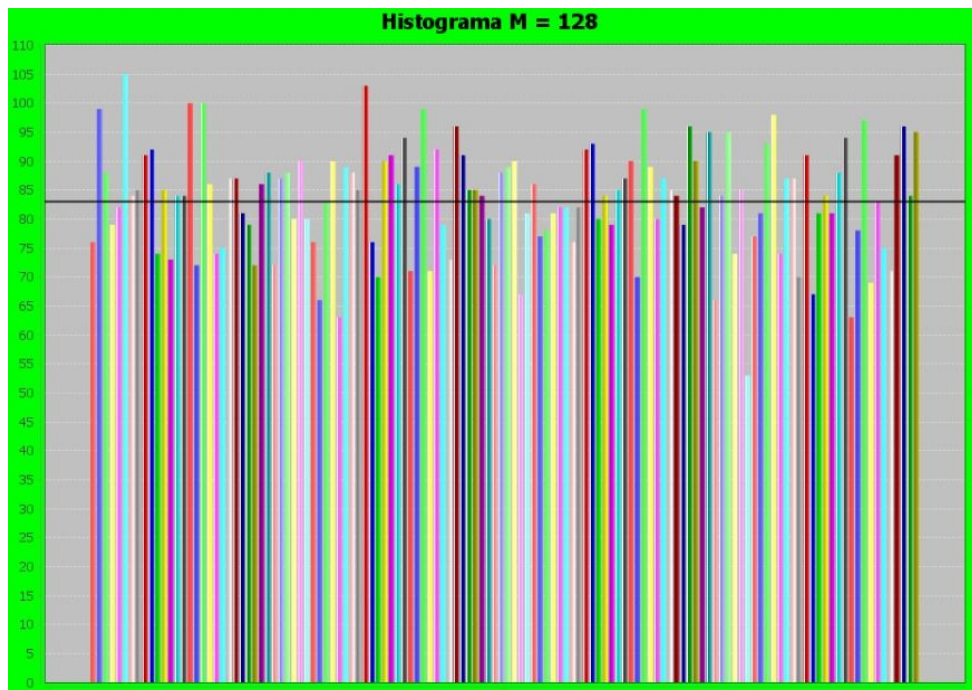
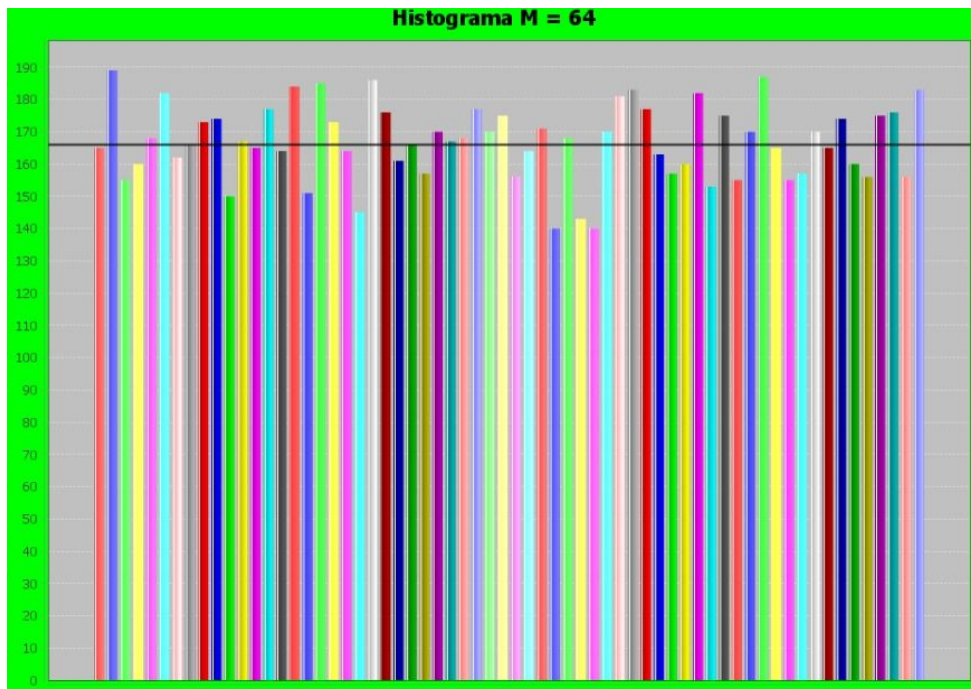


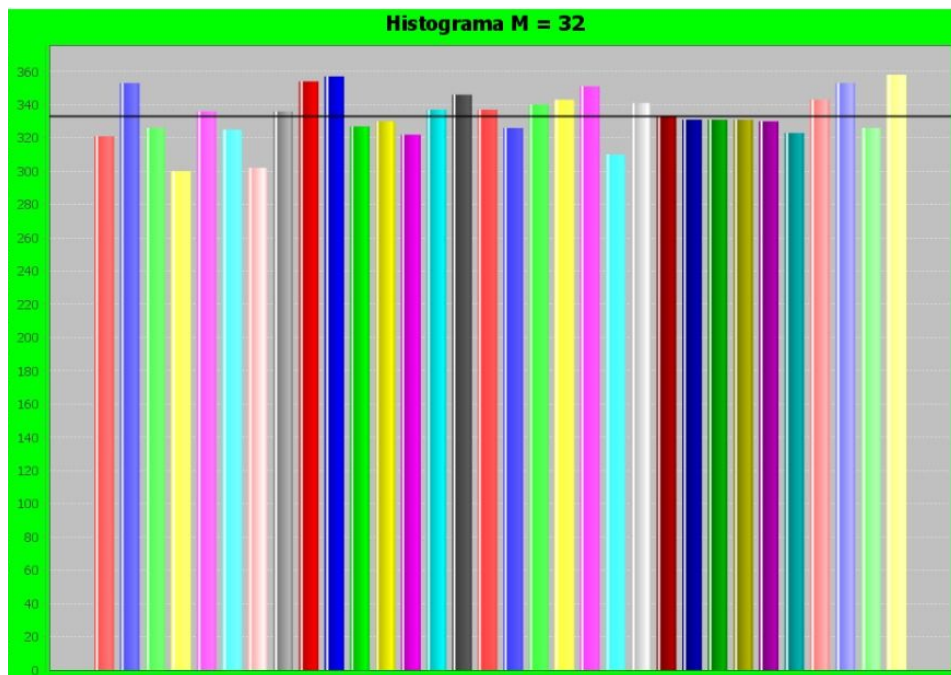
Imagem 3 - Histograma da Tabela hash em que  $M = 128$ .

Usando como  $M$  a potência  $2^6$ :



**Imagem 4 - Histograma da Tabela hash em que  $M = 64$ .**

Usando como  $M$  a potência  $2^4$ :



**Imagem 5 - Histograma da Tabela hash em que  $M = 32$ .**

- 4) Sabendo que  $\alpha = N/M$  é o fator de carga (load factor) de uma Tabela Hash, sendo  $N$  o número de chaves e  $M$  o número de posições da tabela, podemos ter  $\alpha < 1$  numa tabela de hash com encadeamento?

Sim, pois podem haver menos chaves que posições, entretanto isso causa um desperdício de espaço pois haverá posições vagas. Por exemplo, quando  $N = 5$  e  $M = 10$ ,  $\alpha = 0.5$ , e haverá pelo menos 5 espaços vazios.

- 5) Fazer uma aplicação para inserir as chaves E A S Y Q U T I O N, nessa ordem usando hashing com encadeamento, em uma tabela com  $M = 5$  listas. Usar a função de hashing  $11*k \% M$  para transformar a  $k$ -ésima letra do alfabeto em um índice da tabela de hash. Desenhar (manualmente) a estrutura da tabela hash ao final das inserções. Obs: implementar métodos para inserir() e para buscar() determinadas chaves na tabela.
- 6) Descrever as operações min(), max() e dizer se é fácil implementá-las em tabelas de símbolos implementadas com tabelas de hash?

**Observações:**

- Para executar usa-se o cmd (cd diretório □ java -jar nomeJar.jar < tale.txt);
- O valor de M é escolhido pelo cliente ao executar;
- Ambas funções de hashing se encontram no código encaminhado;
- Caso queiram ler o texto e ver no cmd, basta descomentar o código “Iterator” no final do programa e gerar outro JAR;
- O arquivo JAR e o tale.txt estão na mesma pasta para o teste.

**Referências:**

Princeton (2019). *Java Algorithms and Clients*. [online] Disponível em: <https://algs4.cs.princeton.edu/code/> [Acessado em 3 de outubro de 2019].