Algoritmo - *Misra & Gries*

João Amaral - 65772

Resumo **- Este relatório encontra-se dividido em cinco capítulos. Sendo o primeiro capítulo direcionado sobre o trabalho e os objetivos a atingir. O segundo irá conter uma breve explicação sobre o problema do estudo. O terceiro capitulo irá falar sobre o trabalhão desenvolvido focado nos requisitos indicados no enunciado. O quarto capitulo, encontra-se os testes efetuados com o trabalho efetuado. O quinto e ultimo capitulo, contém a conclusão que contém a conclusão dos trabalhos como também as conclusões tirados nos testes.**

Abstract **- This report is divided into five Chapters. Being the first chapter focused on the Work and the objectives to be achieved. The second will contain a brief explanation of the study problem. The third chapter will talk about the work developed focused on the requirements indicated in the statement. The fourth chapter, You will find the tests performed with the work done. The fifth and last chapter contains the conclusion that concludes the work as well as the conclusions drawn in the tests.**

I. INTRODUÇÃO

O trabalho prático realizado e descrito neste relatório pertence ao âmbito da disciplina de Algoritmos Avançados que pertence ao Mestrado de Engenharia de Informática da Universidade de Aveiro.
O trabalho proposto pelo docente da disciplina foi encontrar os itens mais frequentes de um conjunto de dados, explorando métodos que permitem processar conjuntos de dados de grande dimensão utilizando o algoritmo Misra & Gries. Foi desenvolvido um algoritmo que no qual gera ficheiro de texto que contém elementos (letras) minúsculas separadas por um espaço. Dando prioridade a certas letras como pedido no enunciado do trabalho.
Será efetuada uma análise da eficiência computacional e das limitações dos algoritmo desenvolvido tirando as respetivas conclusões.

II. Problema proposto

O problema proposto foi os elementos frequentes num *dataset*. Informalmente é dado uma sequencia de elementos, o problema é simplesmente encontrar os elementos em que ocorrem mais frequentemente.
Normalmente, este problema é mais conhecido como encontrar todos os elementos que a frequência excede uma fração especifica de todos os números dos elementos. (Figura 1).

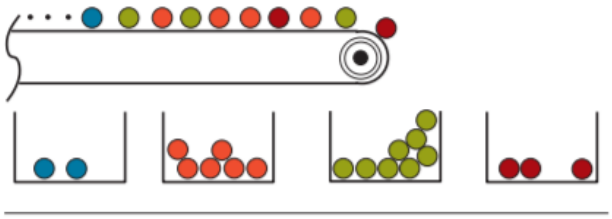

Figura 1 - Um *stream* dedos definindo a frequência

Variações têm tendência de aparecer quando os elementos contêm pesos e mais ainda quando esses pesos forem eventualmente negativos.
Consequente, deste que para guardar corretamente o conjunto de dados de tamanho N requerer um espaço de Ω(N), sendo esse espaço também necessário para resolver o problema de elementos mais frequentes.

III. TRABALHO DESENVOLVIDO

O trabalho foi dividido em 2 partes para que fosse possível corresponder ao que foi pedido pelo docente. A criação de uma ficheiro texto que no qual irá conter letras minúsculas e o algoritmo Misra & Gries.

a. Criação do dataset

Foi criado uma função que através de um alfabeto fosse possível utilizar um método *random* para criar o texto com o numero de elementos pretendidos e indicados pelo utilizador. Como pedido no enunciado, o docente deixou realçado a necessidade de atribuir mais probabilidades a algumas das letras. Para responder a esse requisito simplesmente adicionou-se mais letras no abecedário, como aparece na figura x.

```
string.letters = 'aaabcdeeefghiiijklmnooopqrstuuuvwxyz'
```

Figura 2 - Abecedário

Como podemos ver, ao realçar as vogais [a,e,i,o,u] pelo menos 3 vezes mais que as restantes letras, podemos assim ter a possibilidade de verificar os resultados do algoritmo.

b. Algoritmo Misra & Gries

Para verificar se o elemento guardado é realmente é a *majority* (maioria), uma segunda passagem é necessária para simplesmente contar o verdadeiro numero de ocorrências de um determinado elemento.

```
def freqVerdadeira(dados, returnMisga):
    aux = {}
    returnaux = {}
    for i in dados:
        if i in aux:
            aux[i]+=1
        else:
            aux[i] = 1
    for w in returnMisga:
        for i in aux:
            if w==i:
                returnaux[w] = aux[w]
    return returnaux
```

Figura 3 - Função da contagem completa

Sem esta segunda passagem, o algoritmo tem uma garantia parcial. Se existe um elemento maioritário, é encontrado no fim da primeira passagem, mas o algoritmo e é incapaz de determinar se é o caso.

Um argumento é usado para defender que qualquer elemento que na qual ocorre mais que n/k de vezes, deve ser guardado pelo algoritmo quando termina.

Algorithm 1: FREQUENT(k)

$n \leftarrow 0$;
$T \leftarrow \emptyset$;
foreach i **do**
  $n \leftarrow n+1$;
  if $i \in T$ **then**
    $c_i \leftarrow c_i + 1$;
  else if $|T| < k-1$ **then**
    $T \leftarrow T \cup [i]$;
    $c_i \leftarrow 1$;
  else forall $j \in T$ **do**
    $c_j \leftarrow c_j - 1$;
    if $c_j = 0$ **then** $T \leftarrow T \setminus [j]$;

Figura 4 - Algoritmo Misra & Gries

```
def frequencia(dados, k):
    n = 0  # conta o nº de iterações
    table = {}
    for i in dados:
        n += 1
        if i in table:
            table[i] += 1
        elif len(table) < k - 1:
            table[i] = 1
        else:
            for j in list(table):
                table[j] -= 1
                if table[j] == 0:
                    table.pop(j)
    return table
```

Figura 5 - Função Misra & Gries

IV. TESTE

Os testes efetuados mostram quatro *outputs*, o *array* resultante do algoritmo, a frequência total das letras correspondentes ao *array* do algoritmo e por fim, mostra qual os elementos do *array* algoritmo tem a frequência maior que 5% e 10%.

1) *Dataset* : 100 | k = 10

```
Array de frequencias
{'i': 1, 'e': 2, 'x': 2, 's': 1, 'a': 1, 'o': 1, 'z': 2}
Frequencias totais
{'i': 8, 'e': 11, 'x': 7, 's': 3, 'a': 8, 'o': 7, 'z': 4}
Lista de letras com mais de 10%
['i - 1', 'e - 2', 'x - 2', 's - 1', 'a - 1', 'o - 1', 'z - 2']
Lista de letras com mais de 5%
['i - 1', 'e - 2', 'x - 2', 's - 1', 'a - 1', 'o - 1', 'z - 2']
```

2) *Dataset* : 100 | k = 26

```
Array de frequencias
{'e': 7, 'q': 3, 'f': 1, 'o': 4, 'k': 1, 'v': 3, 'm': 3, 'n': 2, 'r':
 5, 'd': 2, 't': 4, 'i': 6, 'a': 5, 'j': 2, 'p': 2, 'y': 3, 'g': 3,
 'b': 5, 'u': 7, 's': 1, 'h': 1, 'c': 1, 'z': 2, 'x': 1}
Frequencias totais
{'e': 8, 'q': 4, 'f': 2, 'o': 5, 'k': 2, 'v': 4, 'm': 4, 'n': 3, 'r':
 6, 'd': 3, 't': 5, 'i': 7, 'a': 6, 'j': 3, 'p': 3, 'y': 4, 'g': 4,
 'b': 6, 'u': 8, 's': 2, 'h': 2, 'c': 2, 'z': 3, 'x': 2}
Lista de letras com mais de 10%
Não havia letras maior que 10%
Lista de letras com mais de 5%
['e - 7', 'o - 4', 'r - 5', 't - 4', 'i - 6', 'a - 5', 'b - 5', 'u - 7']
```

3) *Dataset* : 100 | k = 50

```
Array de frequencias
{'o': 10, 'w': 2, 'a': 3, 'e': 12, 'c': 3, 'u': 9, 'z': 4, 'g': 5,
 'i': 5, 'd': 3, 'n': 5, 'j': 6, 's': 2, 'p': 3, 'q': 3, 'l': 6, 'x':
 3, 't': 3, 'm': 4, 'b': 1, 'k': 2, 'f': 3, 'y': 1, 'r': 1, 'v': 1}
Frequencias totais
{'o': 10, 'w': 2, 'a': 3, 'e': 12, 'c': 3, 'u': 9, 'z': 4, 'g': 5,
 'i': 5, 'd': 3, 'n': 5, 'j': 6, 's': 2, 'p': 3, 'q': 3, 'l': 6, 'x':
 3, 't': 3, 'm': 4, 'b': 1, 'k': 2, 'f': 3, 'y': 1, 'r': 1, 'v': 1}
Lista de letras com mais de 10%
['o - 10', 'e - 12']
Lista de letras com mais de 5%
['o - 10', 'e - 12', 'u - 9', 'g - 5', 'i - 5', 'n - 5', 'j - 6', 'l -
 6']
```

4) *Dataset* : 1000 | k = 10

```
Array de frequencias
{'o': 4, 'u': 10, 'e': 2, 'w': 1, 'a': 1, 'b': 1, 'v': 1}
Frequencias totais
{'o': 85, 'u': 91, 'e': 83, 'w': 32, 'a': 87, 'b': 30, 'v': 29}
Lista de letras com mais de 10%
['o - 4', 'u - 10', 'e - 2']
Lista de letras com mais de 5%
['o - 4', 'u - 10', 'e - 2', 'w - 1', 'a - 1', 'b - 1', 'v - 1']
```

5) *Dataset* : 1000 | k = 26

```
Array de frequencias
{'r': 14, 'e': 63, 'l': 8, 'c': 14, 'u': 82, 'v': 18, 'a': 77, 'n':
 10, 's': 9, 'i': 65, 'j': 5, 'd': 18, 'h': 10, 'f': 14, 'o': 61, 'z':
 12, 't': 9, 'q': 13, 'k': 6, 'p': 7, 'm': 4, 'w': 6, 'g': 3, 'b': 2,
 'y': 2}
Frequencias totais
{'r': 32, 'e': 81, 'l': 26, 'c': 32, 'u': 100, 'v': 36, 'a': 95, 'n':
 28, 's': 27, 'i': 83, 'j': 23, 'd': 36, 'h': 28, 'f': 32, 'o': 79,
 'z': 30, 't': 27, 'q': 31, 'k': 24, 'p': 25, 'm': 22, 'w': 24, 'g':
 21, 'b': 20, 'y': 20}
Lista de letras com mais de 10%
['e - 63', 'u - 82', 'a - 77', 'i - 65', 'o - 61']
Lista de letras com mais de 5%
['e - 63', 'u - 82', 'a - 77', 'i - 65', 'o - 61']
```

6) *Dataset* : 1000 | k = 50

```
Array de frequencias
{'o': 89, 'a': 86, 'u': 74, 'z': 27, 'j': 33, 'l': 28, 'd': 40, 'i':
 82, 'y': 27, 'x': 35, 'g': 34, 'n': 30, 's': 31, 'm': 34, 'r': 25,
 'f': 23, 'e': 76, 'k': 23, 'c': 29, 'h': 30, 't': 23, 'v': 28, 'q':
 30, 'w': 18, 'b': 25, 'p': 20}
Frequencias totais
{'o': 89, 'a': 86, 'u': 74, 'z': 27, 'j': 33, 'l': 28, 'd': 40, 'i':
 82, 'y': 27, 'x': 35, 'g': 34, 'n': 30, 's': 31, 'm': 34, 'r': 25,
 'f': 23, 'e': 76, 'k': 23, 'c': 29, 'h': 30, 't': 23, 'v': 28, 'q':
 30, 'w': 18, 'b': 25, 'p': 20}
Lista de letras com mais de 10%
Não havia letras maior que 10%
Lista de letras com mais de 5%
['o - 89', 'a - 86', 'u - 74', 'i - 82', 'e - 76']
```

7) *Dataset* : 10000 | k = 10

```
Array de frequencias
{'e': 7, 'a': 14, 'u': 9, 'i': 5, 'l': 1, 'o': 1, 't': 1, 'z': 1, 'w':
 1}
Frequencias totais
{'e': 870, 'a': 844, 'u': 868, 'i': 805, 'l': 273, 'o': 832, 't': 284,
 'z': 281, 'w': 280}
Lista de letras com mais de 10%
['e - 7', 'a - 14', 'u - 9', 'i - 5']
Lista de letras com mais de 5%
['e - 7', 'a - 14', 'u - 9', 'i - 5']
```

8) *Dataset* : 10000 | k = 26

```
Array de frequencias
{'c': 22, 'x': 23, 'h': 45, 'b': 32, 'e': 625, 'i': 626, 'o': 604,
 'u': 603, 'm': 61, 'a': 607, 'g': 43, 'w': 47, 'v': 69, 's': 33, 'q':
 39, 't': 50, 'l': 45, 'j': 29, 'f': 46, 'k': 33, 'p': 25, 'n': 42,
 'r': 58, 'z': 51, 'd': 32}
Frequencias totais
{'c': 257, 'x': 258, 'h': 280, 'b': 267, 'e': 860, 'i': 861, 'o': 839,
 'u': 838, 'm': 296, 'a': 842, 'g': 278, 'w': 282, 'v': 304, 's': 268,
 'q': 274, 't': 285, 'l': 280, 'j': 264, 'f': 281, 'k': 268, 'p': 260,
 'n': 277, 'r': 293, 'z': 286, 'd': 267}
Lista de letras com mais de 10%
['e - 625', 'i - 626', 'o - 604', 'u - 603', 'a - 607']
Lista de letras com mais de 5%
['e - 625', 'i - 626', 'o - 604', 'u - 603', 'a - 607']
```

9) *Dataset* : 10000 | k = 50

```
Array de frequencias
{'l': 254, 'b': 301, 'u': 841, 'p': 279, 'j': 285, 'z': 270, 'i': 810,
 'w': 288, 'e': 823, 't': 284, 's': 303, 'o': 850, 'd': 264, 'm': 285,
 'a': 874, 'q': 275, 'k': 239, 'n': 279, 'v': 286, 'r': 260, 'g': 264,
 'x': 281, 'h': 284, 'f': 266, 'c': 280, 'y': 275}
Frequencias totais
{'l': 254, 'b': 301, 'u': 841, 'p': 279, 'j': 285, 'z': 270, 'i': 810,
 'w': 288, 'e': 823, 't': 284, 's': 303, 'o': 850, 'd': 264, 'm': 285,
 'a': 874, 'q': 275, 'k': 239, 'n': 279, 'v': 286, 'r': 260, 'g': 264,
 'x': 281, 'h': 284, 'f': 266, 'c': 280, 'y': 275}
Lista de letras com mais de 10%
Não havia letras maior que 10%
Lista de letras com mais de 5%
['u - 841', 'i - 810', 'e - 823', 'o - 850', 'a - 874']
```

10) *Dataset* : 100000 | k = 10

```
Array de frequencias
{'a': 8, 'u': 2, 'o': 2, 'q': 1, 'p': 2, 'n': 1, 'k': 1, 'x': 2, 't': 1}
Frequencias totais
{'a': 8458, 'u': 8431, 'o': 8282, 'q': 2788, 'p': 2758, 'n': 2722,
 'k': 2728, 'x': 2776, 't': 2906}
Lista de letras com mais de 10%
['a - 8', 'u - 2', 'o - 2', 'p - 2', 'x - 2']
Lista de letras com mais de 5%
['a - 8', 'u - 2', 'o - 2', 'q - 1', 'p - 2', 'n - 1', 'k - 1', 'x -
 2', 't - 1']
```

11) *Dataset* : 100000 | k = 26

```
Array de frequencias
{'a': 5624, 'e': 5865, 'v': 152, 'o': 5692, 'n': 238, 'w': 193, 'f':
 220, 'i': 5946, 'q': 195, 'u': 5840, 'p': 169, 'j': 196, 'g': 163,
 'b': 145, 'r': 206, 'c': 161, 'm': 138, 'l': 221, 'z': 136, 's': 222,
 'h': 212, 'x': 153, 'd': 127, 'k': 266, 'y': 180}
Frequencias totais
{'a': 8214, 'e': 8455, 'v': 2742, 'o': 8282, 'n': 2828, 'w': 2783,
 'f': 2810, 'i': 8536, 'q': 2785, 'u': 8430, 'p': 2759, 'j': 2786,
 'g': 2753, 'b': 2735, 'r': 2796, 'c': 2751, 'm': 2728, 'l': 2811,
 'z': 2726, 's': 2812, 'h': 2802, 'x': 2743, 'd': 2717, 'k': 2856,
 'y': 2770}
Lista de letras com mais de 10%
['a - 5624', 'e - 5865', 'o - 5692', 'i - 5946', 'u - 5840']
Lista de letras com mais de 5%
['a - 5624', 'e - 5865', 'o - 5692', 'i - 5946', 'u - 5840']
```

12) *Dataset* : 100000 | k = 50

```
Array de frequencias
{'j': 2782, 's': 2668, 'e': 8336, 'g': 2796, 'a': 8242, 'h': 2824,
 'c': 2817, 'u': 8400, 'i': 8207, 'o': 8362, 'b': 2780, 'r': 2815,
 'q': 2812, 'y': 2813, 'z': 2749, 'm': 2835, 'x': 2779, 'f': 2795,
 'w': 2792, 'l': 2730, 'p': 2694, 'n': 2742, 'd': 2854, 't': 2794,
 'k': 2792, 'v': 2790}
Frequencias totais
{'j': 2782, 's': 2668, 'e': 8336, 'g': 2796, 'a': 8242, 'h': 2824,
 'c': 2817, 'u': 8400, 'i': 8207, 'o': 8362, 'b': 2780, 'r': 2815,
 'q': 2812, 'y': 2813, 'z': 2749, 'm': 2835, 'x': 2779, 'f': 2795,
 'w': 2792, 'l': 2730, 'p': 2694, 'n': 2742, 'd': 2854, 't': 2794,
 'k': 2792, 'v': 2790}
Lista de letras com mais de 10%
Não havia letras maior que 10%
Lista de letras com mais de 5%
['e - 8336', 'a - 8242', 'u - 8400', 'i - 8207', 'o - 8362']
```

V. CONCLUSÃO

O que posso concluir ao examinar os testes efetuados ao verificar os diferentes outputs nos testes com diferentes tamanhos dos *datasets* com diferentes valores de k, éo facto da eventualidade do k ser pequeno irá originar valores parcialmente corretos, indicando apenas os elementos que têm uma maior frequência. Cujo problema é diluído quando se aumenta o valor do k. Devolvendo assim a lista da frequência de cada elemento. Tendo em conta que a gerada pelo algoritmo respeita a limitação de (k-1).

REFERENCIAS

[1] Apontamentos do docente da disciplina.

[2] Cormode G., Hadjieleftheriou M. Finding the Frequent Items in Streams of Data, *Communications of the ACM*,*52*, 2009, 97-105.