



Centro Universitário FEI

Projeto de CA-2330

Relatório - Parte 2

Grupo Nº 11

João Pedro Rosa Cezarino - R.A: 22.120.021-5

Lucca Bonsi Guarreschi - R.A: 22.120.016-5

Vitor Martins Oliveira - R.A: 22.120.067-8

**São Bernardo do Campo
2020**

1 Graus de um grafo

O grau de um vértice é dado pelo número de arestas que lhe são incidentes, com laços contados duas vezes.

Em G, por exemplo:

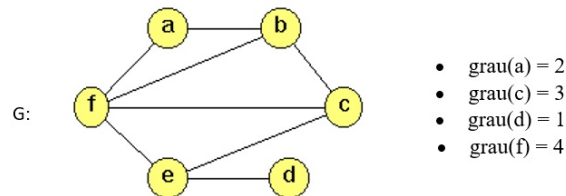


Figura 1

A Sequência de graus de um grafo consiste em escrever em ordem crescente ou decrescente os graus de seus vértices.

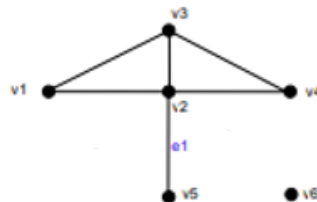


Figura 2: Sequência de graus = 4, 3, 2, 2, 1, 0

A soma dos graus de todas as arestas de um grafo é duas vezes o número de arestas, já que cada aresta contribui para dois graus.

$$\sum_{i=1}^n d(V_i) = 2a$$

Um vértice que possui grau zero é um vértice isolado.

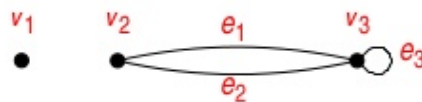


Figura 3: V1 é um vértice isolado grau(1) = 0

Se o grafo não conter nenhuma aresta, então todos os vértices são isolados e o grafo é chamado grafo nulo.

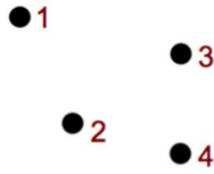


Figura 4

2 Arestas de um grafo

Uma aresta é um par não-ordenado (v_i, v_j) , onde v_i e v_j são elementos de V .

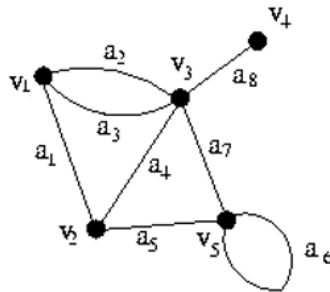


Figura 5: $E = a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8$

Arestas Adjacentes são duas arestas com um extremo em comum.

Arestas Múltiplas são arestas que possuem os mesmos extremos.

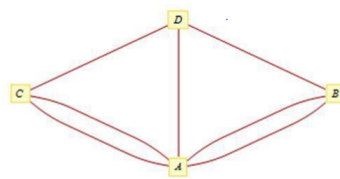


Figura 6

3 Grafos Completos

Um grafo completo com v vértices é um grafo simples onde todo par de vértices é ligado por uma aresta. Logo, um grafo completo é um grafo simples que contém o número máximo de arestas. Usualmente denota-se esse grafo por K_n , onde n é a ordem do grafo.

Um grafo Completo também é regular($n - 1$), pois todos os seus vértices têm grau $n - 1$.

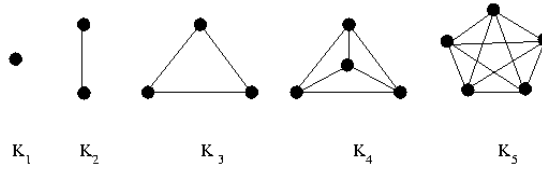


Figura 7

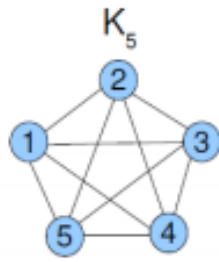
Para se calcular o número de arestas de um grafo completo, utiliza-se a fórmula:

$$\frac{n(n-1)}{2}$$

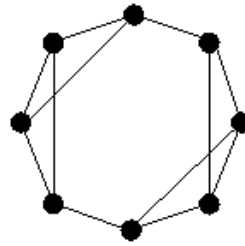
n = número de vértices do grafo

4 Grafos Regulares

Um grafo regular é um grafo onde cada vértice tem o mesmo número de arestas adjacentes, ou seja, cada vértice tem o mesmo grau.



(a) Grafo regular de grau 4



(b) Grafo regular de grau 3

5 Descrição do Programa

Inicialmente o arquivo onde se encontra a matriz de adjacência "A.txt" é aberto e a leitura de cada linha da matriz é realizada, com o objetivo de retirar os espaços (" \n") e transformar o conteúdo em números inteiros (do tipo "int"). Após esse processo, o programa verifica e retira (se houver a presença) os espaços em branco desnecessários na matriz.

```

arq = open('B.txt', 'r') ## Abre o arquivo no modo "read" ##
lista = arq.readlines() ## Lê o arquivo "B.txt" e cria uma lista com seu conteúdo ##
lista_final = []
arq.close()

for x in lista: ## Esse "for" lê a lista e o arquivo e adiciona o seus elementos a outra lista, removendo as quebras de linhas ##
    x = x.rstrip('\n')
    lista_final.append(x)

for num in lista_final: ## Remove os espaços em branco do final da lista, caso tenha algum ##
    if '' in lista_final:
        lista_final.remove('')

Matriz = [[int(num) for num in line.split(' ')] for line in lista_final] ## Transforma a lista em uma matriz ##
Matriz2 = [[int(num) for num in line.split(' ')] for line in lista_final]

```

Figura 9

Logo após a execução deste primeiro bloco de código, são criadas as variáveis "arestas" (a qual é igualada a zero) e uma lista com o nome "grau", que será responsável por armazenar a sequência dos graus do grafo em questão. Com isso, um laço de repetição é criado para percorrer toda a extensão da matriz (suas linhas e colunas) e caso o laço encontre um valor maior ou igual a 1 quando i e j da matriz a_{ij} forem iguais então, o valor é dobrado (pois um laço conta como 2 graus no vértice) e adicionado a lista "grau".

```

16  ## Faz a Sequência dos graus ##
17  arestas = 0
18  grau = []
19
20  ## Faz a Sequência dos graus ##
21  for linha in range(len(Matriz2)):
22      for coluna in range(len(Matriz2[linha])):
23          if coluna == linha and Matriz2[linha][coluna] >= 1:
24              Matriz2[linha][coluna] = Matriz2[linha][coluna]*2
25  for linha in Matriz2:
26      grau.append(sum(linha))
27

```

Figura 10

Após o armazenamento da sequência gráfica, outro laço de repetição é criado, com o intuito de contar o número de arestas do grafo e imprimi-lo.

Para verificar se o grafo é completo ou não, utilizam-se duas novas variáveis também igualadas a zero, "aresta_multipla" e "laco". A verificação é feita por meio da presença de arestas múltiplas e laços. Caso o grafo contenha arestas múltiplas ou laços ele deixa de ser um grafo simples, logo, não pode ser considerado completo.

No entanto, se o grafo for simples (não possuir arestas múltiplas nem laços) parte-se para outra condição: Os graus de cada vértice são analisados e comparados uns aos outros, caso os graus dos vértices forem iguais, conclui-se que o grafo é completo. Porém, se os graus dos vértices não forem iguais, então, o grafo não é considerado completo

```

28  ## Conta o número de arestas ##
29  for linha in range(len(Matriz2)):
30      for coluna in range(len(Matriz2[linha])):
31          arestas = arestas + Matriz2[linha][coluna]
32
33  print("\n***** RESULTADOS *****\n")
34  print(f"O grafo possui {int(arestas/2)} arestas\n")
35
36  ## Verifica se o grafo é completo ##
37  aresta_multipla = 0
38  laco = 0
39
40  for linha in range(len(Matriz)):
41      for coluna in range(len(Matriz[linha])):
42          if linha == coluna and Matriz[linha][coluna] > 0:
43              laco = laco + Matriz[linha][coluna]
44          if Matriz[linha][coluna] > 1 and linha != coluna:
45              aresta_multipla = aresta_multipla + 1

```

Figura 11

```

47  ## Verifica se o grafo possui laços ou arestas múltiplas ##
48  if laço > 0 or aresta_multipla > 1:
49      print(f"* 0 grafo não é completo!\n")
50  elif laço == 0 and aresta_multipla == 0:
51      verifica = 0
52      ant = 0
53      pro = 0
54      for x in grau:
55          pro = x
56          if x == ant:
57              verifica = 1
58          else:
59              ant = pro
60              verifica = 0
61
62      if verifica == 0:
63          print(f"* 0 grafo não é completo!\n")
64      elif verifica == 1:
65          print(f"* 0 grafo é completo!\n")
66
67
68

```

Figura 12

Um grafo é considerado regular quando todos os seus vértices possuem o mesmo grau. Para verificar tal condição utilizou-se um laço de repetição para percorrer a lista "grau". Nele, os graus são comparados uns com os outros e caso todos graus dos vértices forem iguais, o grafo é considerado regular. Do contrário, o grafo é considerado não regular.

Por fim, a sequência dos graus do grafo é colocado em ordem decrescente por meio da função ".sort" e do argumento "reverse".

```

70  ## Verifica se o grafo é regular ##
71  verifica2 = 0
72  ant = 0
73  pro = 0
74  for x in grau:
75      pro = x
76      if x == ant:
77          verifica2 = 1
78      else:
79          ant = pro
80          verifica2 = 0
81
82  if verifica2 == 0:
83      print(f"* 0 grafo não é regular!\n")
84  elif verifica2 == 1:
85      print(f"* 0 grafo é regular!\n")
86
87
88  ## Coloca a sequência dos graus em ordem decrescente ##
89  grau.sort(reverse=True)
90
91  print(f"* Sequência de graus do grafo = {grau}")
92

```

Figura 13

Após a execução completa de todos os blocos de código representados e descritos acima, as respostas à todas as questões propostas são impressas no terminal da seguinte maneira:

```
***** RESULTADOS *****

* O grafo possui 20 arestas

* O grafo não é completo!

* O grafo não é regular!

* Sequência de graus do grafo = [7, 5, 4, 4, 3, 3, 3, 3, 3, 3, 2]

*****

----- INTEGRANTES -----
-----
Nome: Lucca Bonsi Guarreschi / R.A:22.120.016-5
-----
Nome: Vitor Martins Oliveira / R.A:22.120.067-8
-----
Nome: João Pedro Rosa Cezarino / R.A: 22.120.021-5
-----
Turma: 010
-----
```

Figura 14