

Atividade 1: Componentes f-conexas

Nome : Messias Feres Curi Melo

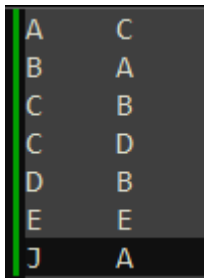
Matrícula : 2022003764

Durante o desenvolvimento do código, testei varias funções das biblioteca networkx e existe até um comando “nx.strongly_connected_components” que resolveria todos os problemas, mas como não pode usar, quebrei um pouco a cabeça até chegar na solução.

O código que desenvolvi faz a leitura de um grafo de um arquivo .txt chamado grafos.txt, e através dessa leitura é adicionado os valores em uma variável dicionário no python. Com os dados já definidos, foi realizado a busca pelos componentes fortemente conexas, para isso, tirei ideia dos exemplos dos slides do professor, mas acabei fazendo um novo caminho, utilizando duas funções separadas inicialmente, uma para buscar o fecho transitivo direto e outra para o transitivo inverso, com essas duas funções prontas, foi a parte de integrar elas na função da busca dos componentes fortemente conexas, assim, fiz um ‘for’ para puxar a função do transitivo direto e me retornar uma pilha com seus vértices e em seguida fiz meus dados do grafo serem transpostos, de traz para frente para poderem ser usados na próxima etapa. Logo depois, foi realizado um ‘while’ para finalmente definir meus componentes f-conexas, então, foi feito verificações e utilizou o fecho transitivo inverso para fazer o caminho inverso e identificar os componentes, e por fim, os resultados foram adicionados em uma lista e exibido no console.

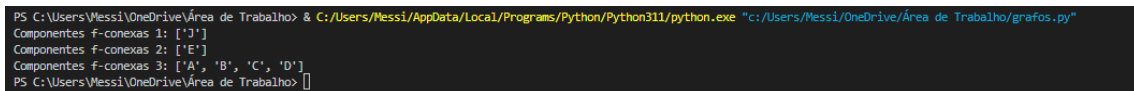
Foi realizado o teste com a função “nx.strongly_connected_components” para conferir se o resultado obtido condizia com o esperado, e foi o mesmo resultado do código desenvolvido durante a atividade.

Entrada:



A	C
B	A
C	B
C	D
D	B
E	E
J	A

Saída:



```
PS C:\Users\Messi\OneDrive\Área de Trabalho> & C:/Users/Messi/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Messi/OneDrive/Área de Trabalho/grafos.py"
Componentes f-conexas 1: ['J']
Componentes f-conexas 2: ['E']
Componentes f-conexas 3: ['A', 'B', 'C', 'D']
PS C:\Users\Messi\OneDrive\Área de Trabalho>
```

Código:

```
C:\Users\Messi> OneDrive > Área de Trabalho > grafos.py > fechoTransitivoDireto

1 def lerGrafo(txt):
2     grafo = {}
3     with open(txt, 'r') as file:
4         for linha in file:
5             u, v = linha.strip().split()
6             if u not in grafo:
7                 grafo[u] = []
8             grafo[u].append(v)
9     return grafo
10
11 def fechoTransitivoDireto(grafo, vertice, visitados):
12     visitados.add(vertice)
13     if vertice in grafo:
14         for vizinho in grafo[vertice]:
15             if vizinho not in visitados:
16                 fechoTransitivoDireto(grafo, vizinho, visitados)
17
18 def fechoTransitivoInverso(grafo, vertice, visitados, componenteAtual):
19     visitados.add(vertice)
20     componenteAtual.append(vertice)
21     if vertice in grafo:
22         for vizinho in grafo[vertice]:
23             if vizinho not in visitados:
24                 fechoTransitivoInverso(grafo, vizinho, visitados, componenteAtual)
25
26 def componentesFortementeConexas(grafo):
27     visitados = set()
28     pilha = []
29     for vertice in grafo.keys():
30         if vertice not in visitados:
31             fechoTransitivoDireto(grafo, vertice, visitados)
32             pilha.append(vertice)
33     grafoTransposto = {}
34     for u in grafo:
35         for y in grafo[u]:
36             if y not in grafoTransposto:
37                 grafoTransposto[y] = []
38             grafoTransposto[y].append(u)
39     visitados.clear()
40     listaFortementeConexas = []
41     while pilha:
42         vertice = pilha.pop()
43         if vertice not in visitados:
44             componenteAtual = []
45             fechoTransitivoInverso(grafoTransposto, vertice, visitados, componenteAtual)
46             listaFortementeConexas.append(componenteAtual)
47
48     return listaFortementeConexas
49
50 def imprimirResultado(componentes):
51     for i, componente in enumerate(componentes):
52         print(f"Componentes f-conexas {i + 1}: {componente}")
53
54 if __name__ == "__main__":
55     arquivo_grafo = "grafos.txt"
56     grafo = lerGrafo(arquivo_grafo)
57     resultados = componentesFortementeConexas(grafo)
58     imprimirResultado(resultados)
```