

1 Instalação

Os comandos da listagem 1 devem ser executados no terminal do Ubuntu para a instalação do software Graphviz e da interface do Python para o Graphviz.

```
1 sudo apt install -y python-pip graphviz libgraphviz-dev pkg-config
2 sudo pip3 install pygraphviz
```

Listagem 1: Instalação

2 Criação do Grafo

Para importar o pacote em um programa Python você deve usar:

```
1 import pygraphviz as pgv
```

Listagem 2: Importação

Para inicializar um grafo:

```
1 G=pgv.AGraph()
```

Listagem 3: Inicialização de um grafo

Para inicializar um dígrafo:

```
1 G=pgv.AGraph(directed=True)
```

Listagem 4: Inicialização de um dígrafo

Para adicionar um vértice:

```
1 G.add_node('o')
```

Listagem 5: Adiciona o vértice “o” ao grafo.

Para adicionar uma aresta entre dois vértices:

```
1 G.add_edge('o','t') # Adiciona uma aresta entre os vértices 'o' e 't'
```

Listagem 6: Adiciona uma aresta entre os vértices 'o' e 't'.

3 Arquivos com Grafos

Salvando o grafo em um arquivo “.dot”:

```
1 G.write("meugrafo.dot")
```

Listagem 7: Salva o grafo em um arquivo.

Criando um grafo a partir de um arquivo salvo previamente:

```
1 G=pgv.AGraph("meugrafo.dot")
```

Listagem 8: Inicializa um grafo a partir de um arquivo salvo previamente.

4 Desenhando o Grafo

O primeiro passo é utilizar um algoritmo de *layout* para posicionar os vértices e arestas. Este processo envolve o uso de software do pacote graphviz.

```
1 G.layout() # Usa o algoritmo implementado no programa neato
2 G.layout(prog='dot') # Usa o algoritmo implementado no programa dot
```

Listagem 9: Posiciona os vértices do grafo.

Depois que o grafo for posicionado é possível gerar um arquivo com uma imagem do grafo.

```
1 G.draw("meugrafo.png") # Salva a imagem em formato png
2 G.draw("meugrafo.ps", prog='circo') # Salva a imagem no formato Post
  ↪ Script
```

Listagem 10: Gera imagens do grafo.

5 Gerando Imagens dos Grafos

O programa da listagem 11 gera uma imagem a partir do grafo informado.

```

1  import pygraphviz as pgv
2
3  def salvaImagemGrafo(meuGrafo, arquivo):
4      G = pgv.AGraph()
5      for u in meuGrafo:
6          G.add_node(u)
7
8      for u in meuGrafo:
9          for v in meuGrafo[u]['adjacencias']:
10             G.add_edge(u, v)
11      G.layout(prog='dot')
12      G.draw(arquivo)
13
14  # Constrói o grafo
15  G = {}
16  #Leia ate não ter mais nada para ler
17  linha=input()
18  while(linha != ""):
19      v1,v2=linha.split()
20      if(v1 not in G):
21          G[v1]={ 'cor':'BRANCO', 'd':0, 'f':0, 'pai':'NULO',
22                  ↪ 'adjacencias':list([v2])}
23      else:
24          G[v1]['adjacencias'].append(v2)
25      if(v2 not in G):
26          G[v2]={ 'cor':'BRANCO', 'd':0, 'f':0, 'pai':'NULO',
27                  ↪ 'adjacencias':list([v1])}
28      else:
29          G[v2]['adjacencias'].append(v1)
30      linha=input()
31
32  #Chegou aqui o grafo esta montado.
33  salvaImagemGrafo(G, "meugrafo.png")

```

Listagem 11: Salva a imagem de um grafo.

6 Exercício

Neste exercício você deve escrever um programa utilitário para manipular grafos. Seu programa deve ser capaz de ler grafos e dígrafos de um arquivo “.dot”, salvar grafos e dígrafos em arquivos “.dot”, gerar imagens a partir dos grafos/dígrafos e, finalmente, deve suportar a entrada de dois ou um vértice por linha.