

IE-0217 Estructuras abstractas de datos y algoritmos para ingeniería

Lab6: Grafos Python

Timna Belinda Brown Ramírez

B61254

`timna.brown@ucr.ac.cr`

`belindabrownr@gmail.com`

I-2019

Tabla de contenidos

1. Enunciado	2
2. Consideraciones	2
3. Abordaje y conclusiones	2
4. Apéndice	3
4.1. Código fuente	3

1. Enunciado

Mapas (gráfos)

Usando: <https://gitlab.com/rica01/ejemplos/tree/master/graph>:

1. modifique el código para que acepte aristas dirigidas.
2. Implementar 3 de los siguientes algoritmos:

- búsqueda en ancho y búsqueda en profundidad
- Dijkstra (camino más corto)
- Prim (AAM)
- Kruskal (AAM)
- puntos de articulación

Posted Mon Jun 17, 2019 at 10:29 am

2. Consideraciones

- Haga grupos de hasta 3 personas.
- Genere un reporte en \LaTeX que incluya su código, su abordaje para la solución y sus conclusiones.
- Suba su código y documentación (doxygen, README, INSTALL) al git respectivo de su grupo y el directorio del laboratorio.
- Cada estudiante debe subir el reporte a Schoology.
- Recuerde que por cada día tardío de entrega se le rebajaran puntos de acuerdo con la formula: 4^d , donde $d > 1$ es la cantidad de días tardíos.

3. Abordaje y conclusiones

Para la resolución del laboratorio presentado, se realizaron una serie de clases y funciones que cumplen con los objetivos del Lab6. El código se encuentra en el gitlab establecido bajo el nombre Lab6-B61254.

Como conclusión se puso en práctica el uso del lenguaje Python, además, del uso de la lógica para cumplir el objetivo planteado.[4]

4. Apéndice

4.1. Código fuente

[3]

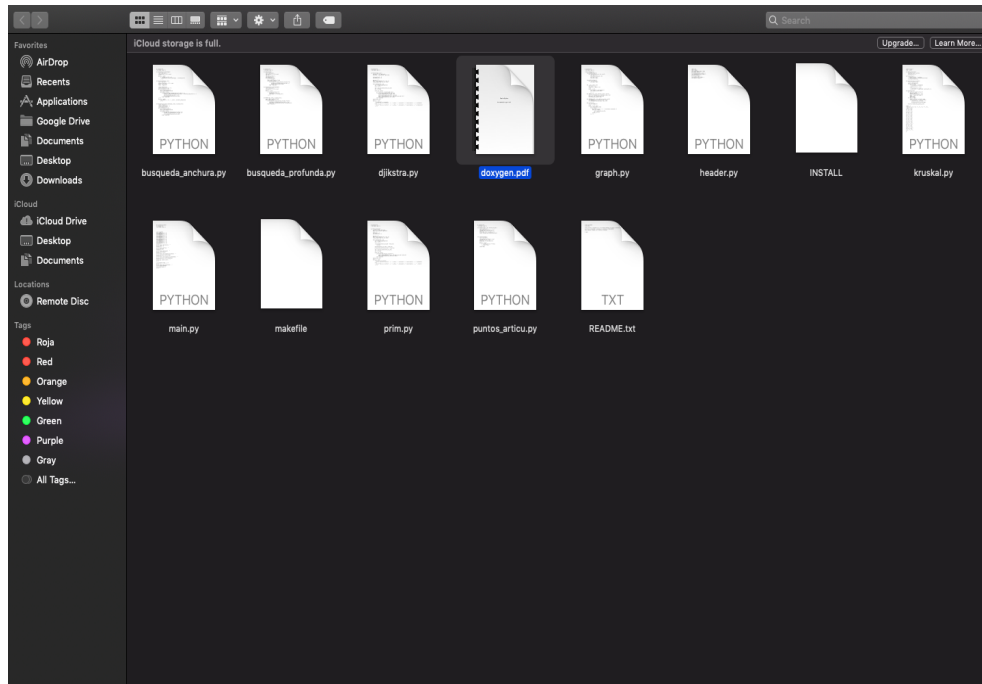


Figura 1: Dentro de Lab6-B61254

Todos tienen en común lo siguiente:

```
Grafos en Python
Julio 2019
```

```
Para ejecutar el programa es ir a la carpeta mediante su consola o
terminal al folder
en el que se encuentra el programa, la carpeta de "Lab6-B61254".
Debe ingresar y digitar en la consola o terminal:
```

```
$ make
```

Figura 2: Readme[2]

Grafos en Python

License Apache 2.0

Se distribuye dos atributos Makefile:

- * build: interpreta el código.
- * clean: limpia los ejecutables.

Figura 3: Install

```
all: build clean

build:
    python main.py

clean:
    rm *.pyc
```

Figura 4: Makefile[1]

```
header.py
1 import sys
2 import heapq
3 from collections import deque
4 from graph import *
5 from busqueda_anchura import *
6 from busqueda_profunda import *
7 from prim import *
8 from kruskal import *
9 from puntos_articu import *
10 from djikstra import *
11
```

Figura 5: Código de header.py

```

Belindas-MacBook-Air:Lab6-B61254 belindabrown$ make
python main.py

Tester de Dijkstra:
Este es el vertice 0 ...antes... 2 a una distancia → → 6
Este es el vertice 1 ...antes... -1 a una distancia → → 0
Este es el vertice 2 ...antes... 1 a una distancia → → 5
Este es el vertice 3 ...antes... 1 a una distancia → → 7
Este es el vertice 4 ...antes... 1 a una distancia → → 9
Este es el vertice 5 ...antes... 3 a una distancia → → 11
Este es el vertice 6 ...antes... 4 a una distancia → → 12
Este es el vertice 7 ...antes... 5 a una distancia → → 12
Este es el vertice 8 ...antes... 7 a una distancia → → 29

Tester de Prim:
Este es el vertice 0 ...antes... 2 a una distancia → → → 1
Este es el vertice 1 ...antes... 1 a una distancia → → → 0
Este es el vertice 2 ...antes... 1 a una distancia → → → 5
Este es el vertice 3 ...antes... 2 a una distancia → → → 2
Este es el vertice 4 ...antes... 5 a una distancia → → → 6
Este es el vertice 5 ...antes... 3 a una distancia → → → 4
Este es el vertice 6 ...antes... 4 a una distancia → → → 3
Este es el vertice 7 ...antes... 5 a una distancia → → → 1
Este es el vertice 8 ...antes... 7 a una distancia → → → 17

Tester de Kruskal:
[(5, 'A', 'D'), (5, 'C', 'E'), (6, 'D', 'F'), (7, 'A', 'B'), (7, 'B', 'E'), (9, 'E', 'G')]

Tester de busqueda por anchura:
None

Tester de busqueda por profundidad:
None

¿El camino existe?:

True

Puntos de articulación:
True
2 a 0 para 2
-1 a 1 para 0
1 a 2 para 1
1 a 3 para 1
1 a 4 para 1
4 a 5 para 2
4 a 6 para 2
6 a 7 para 3
7 a 8 para 4

rm *.pyc
Belindas-MacBook-Air:Lab6-B61254 belindabrown$ █

```

Figura 6: Resultados

Referencias

- [1] Kroah-Hartman G Corbet. J, Rubini. A. *Linux Coding*. O'Reilly books, 1998.
- [2] Computer Science Labs. *Tecnology- commands*. O'Reilly books, 2018.
- [3] Mark Summerfield. *Programming in Python 3: A Complete Introduction to the Python Language*. Anaya Multimedia, 2009.
- [4] A. M. Turing. *On computable numbers with an application to the Entscheidungs problem*. Proceedings of the london mathematical society, 1997.