



Código Asignatura:
ISC-314

Nombre:
Ronald Mariotti

Matricula:
2014-0698

Trabajo:
Tarea 6 - Corte Mínimo

README

En esta tarea lo que se pide es realizar un programa de los cortes mínimos en un grafo con el algoritmo de Karger que se aprendió en clase, se va usar el archivo kargerMinCut.txt con 200 vértices, en este trabajo lo que se busca es crear y ejecutar el código que está hecho en Java para calcular el corte mínimo del archivo, por lo que se explica en el problema hay que tener en cuenta que se debe implementar las contracciones de aristas inicialmente para que pueda ser eficiente. Con mi programa he realizado un total de 3 corridas para asegurar el mejor resultado posible que se pueda obtener.

Notas del problema:

- Encontrar dos puntos en un grafo para cortar en ellos, y hacer crear dos subgrafos de manera que la cantidad sea la mínima posible.
- Tener en cuenta que la cantidad de cortes a probar corra en tiempo exponencial con la cantidad de aristas m que tenga el grafo.

Resultados del programa

Primer corrida:

Corte min 22

-

Segunda corrida:

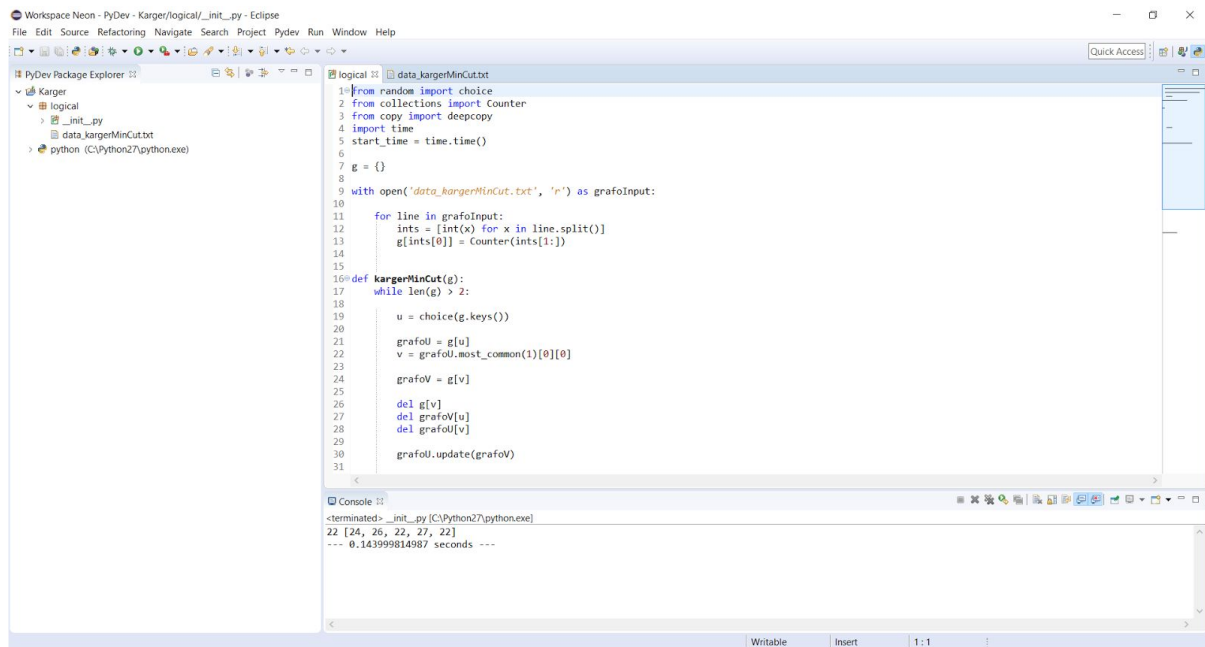
Corte min 20

-

Tercera corrida:

Corte min 17

Captura primer corrida:



Workspace Neon - PyDev - Karger/Logical/_init_.py - Eclipse

File Edit Source Refactoring Navigate Search Project Pydev Run Window Help

PyDev Package Explorer

- Karger
 - logical
 - _init_.py
 - data_kargerMinCut.txt
 - python (C:\Python27\python.exe)

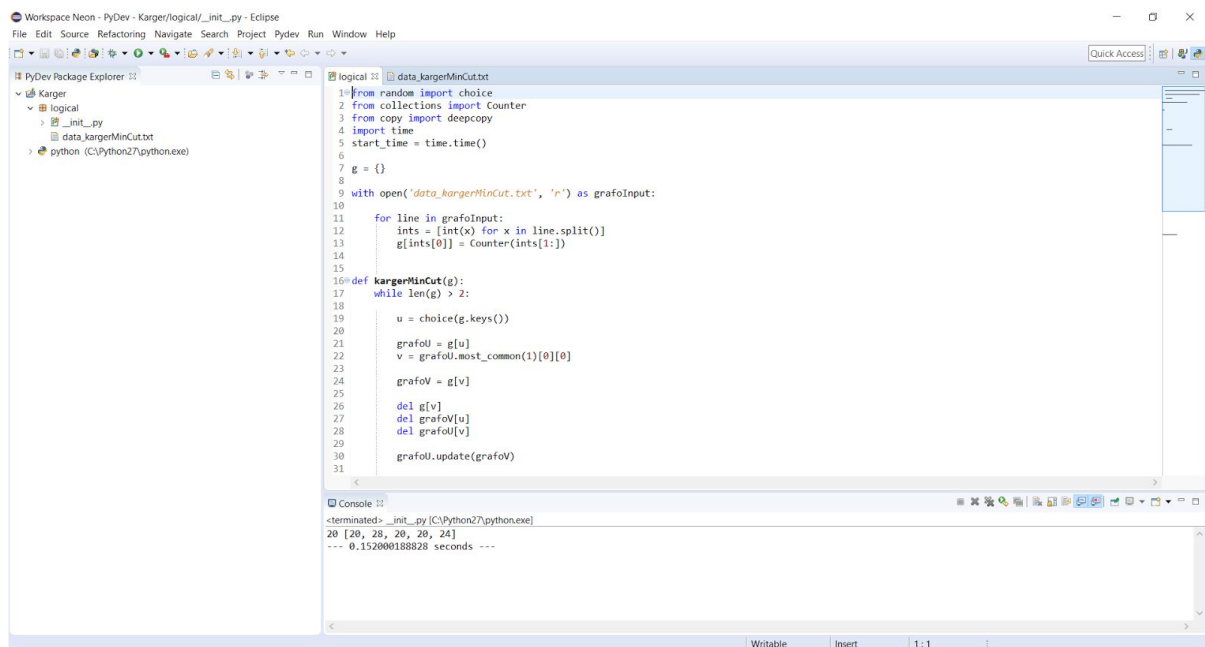
logical :: data_kargerMinCut.txt

```
1 from random import choice
2 from collections import Counter
3 from copy import deepcopy
4 import time
5 start_time = time.time()
6
7 g = {}
8
9 with open('data_kargerMinCut.txt', 'r') as grafoInput:
10     for line in grafoInput:
11         ints = [int(x) for x in line.split()]
12         g[ints[0]] = Counter(ints[1:])
13
14
15
16 def kargerMinCut(g):
17     while len(g) > 2:
18         u = choice(g.keys())
19
20         grafoU = g[u]
21         v = grafoU.most_common(1)[0][0]
22
23         grafoV = g[v]
24
25         del g[v]
26         del grafoU[u]
27         del grafoU[v]
28
29         grafoU.update(grafoV)
30
31
```

Console

```
<terminated> _init_.py [C:\Python27\python.exe]
22 [24, 26, 22, 27, 22]
--- 0.143999814987 seconds ---
```

Captura segunda corrida:



Workspace Neon - PyDev - Karger/Logical/_init_.py - Eclipse

File Edit Source Refactoring Navigate Search Project Pydev Run Window Help

PyDev Package Explorer

- Karger
 - logical
 - _init_.py
 - data_kargerMinCut.txt
 - python (C:\Python27\python.exe)

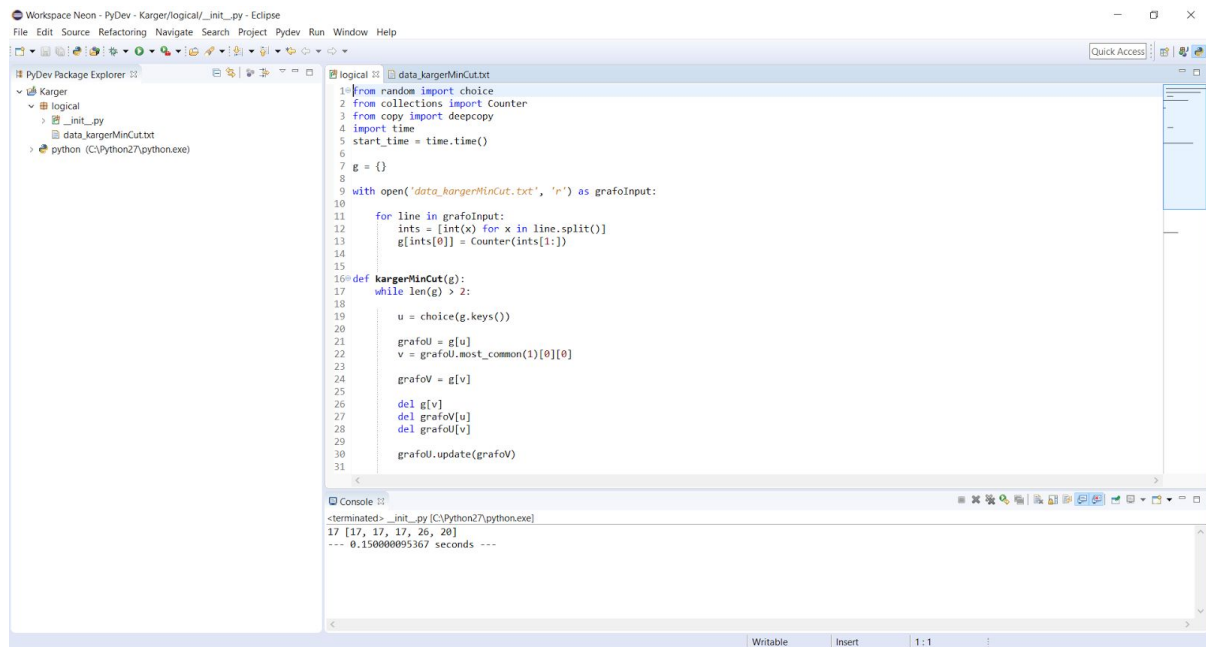
logical :: data_kargerMinCut.txt

```
1 from random import choice
2 from collections import Counter
3 from copy import deepcopy
4 import time
5 start_time = time.time()
6
7 g = {}
8
9 with open('data_kargerMinCut.txt', 'r') as grafoInput:
10     for line in grafoInput:
11         ints = [int(x) for x in line.split()]
12         g[ints[0]] = Counter(ints[1:])
13
14
15
16 def kargerMinCut(g):
17     while len(g) > 2:
18         u = choice(g.keys())
19
20         grafoU = g[u]
21         v = grafoU.most_common(1)[0][0]
22
23         grafoV = g[v]
24
25         del g[v]
26         del grafoU[u]
27         del grafoU[v]
28
29         grafoU.update(grafoV)
30
31
```

Console

```
<terminated> _init_.py [C:\Python27\python.exe]
26 [28, 28, 20, 20, 24]
--- 0.152000188828 seconds ---
```

Captura tercer corrida:



The screenshot shows the Eclipse IDE interface. The main editor displays a Python script named `data_kargerMinCut.txt`. The script implements a Karger's algorithm for finding a minimum cut in a graph. It starts by importing `random`, `choice`, `Counter`, `deepcopy`, and `time`. It then reads a graph from `data_kargerMinCut.txt` and defines a function `kargerMinCut(g)` that repeatedly selects two random nodes, finds their common neighbors, and removes them until only two nodes remain. The console output shows the execution of the script, which terminated successfully and printed the result `[17, 17, 17, 26, 20]` and the execution time `0.150000095367 seconds`.

```
1 from random import choice
2 from collections import Counter
3 from copy import deepcopy
4 import time
5 start_time = time.time()
6
7 g = {}
8
9 with open('data_kargerMinCut.txt', 'r') as grafoInput:
10
11     for line in grafoInput:
12         ints = [int(x) for x in line.split()]
13         g[ints[0]] = Counter(ints[1:])
14
15
16 def kargerMinCut(g):
17     while len(g) > 2:
18
19         u = choice(g.keys())
20
21         grafoU = g[u]
22         v = grafoU.most_common(1)[0][0]
23
24         grafoV = g[v]
25
26         del g[v]
27         del grafoU[u]
28         del grafoU[v]
29
30         grafoU.update(grafoV)
31
```

Console Output:

```
<terminated> _init_py [C:\Python27\python.exe]
17 [17, 17, 17, 26, 20]
--- 0.150000095367 seconds ---
```