

Ejecución de los tres algoritmos

Todo se ejecuta en la pantalla de la consola de Python, yo he utilizado ipython para la ejecución de los algoritmos.

Algoritmo BT

Se importa el modulo desde la ubicación del archivo,

- 1.- Declara un arreglo de puntos vacío, pts=[],
- 2.-ingresa el número de vértices del grafo n=3
- 3.-Se ingresa el número de cruces a calcular de este grafo, este valor le dará una de las condiciones de paro, dentro del archivo se encuentra el número de iteraciones a realizar
- 4.- se manda a llamar a la ejecución como se muestra en la línea final, la salida será una terna donde las posiciones son respectivamente (número de cruces obtenido, numero de vértices, tamaño de la vecindad de búsqueda)

```
In [11]:  
In [11]: import A_BT  
In [12]: pts=[]  
In [13]: n=3  
In [14]: min_val=0  
In [15]: A_BT.tries_lt2(n,pts=pts,min_val=min_val)  
Out[15]: (0, 3, 100000000)
```

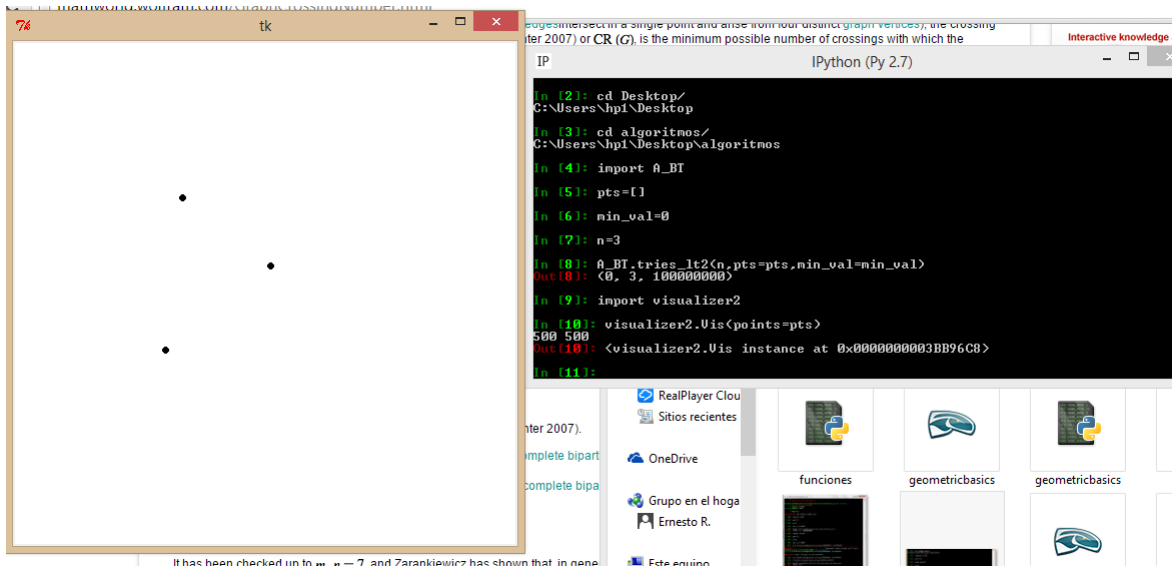
Para usar el visualizador de puntos

Al final de la ejecución se importa visualizer2, como se observa en la línea 9

Posteriormente se ingresa como argumento el arreglo de puntos línea 10

```
In [31]: cd algoritmos/  
C:\Users\hpl\Desktop\algoritmos  
In [4]: import A_BT  
In [5]: pts=[]  
In [6]: min_val=0  
In [7]: n=3  
In [8]: A_BT.tries_lt2(n,pts=pts,min_val=min_val)  
Out[8]: (0, 3, 100000000)  
In [9]: import visualizer2  
In [10]: visualizer2.Vis(points=pts)  
500 500  
Out[10]: <visualizer2.Vis instance at 0x0000000003BB96C8>  
In [11]:
```

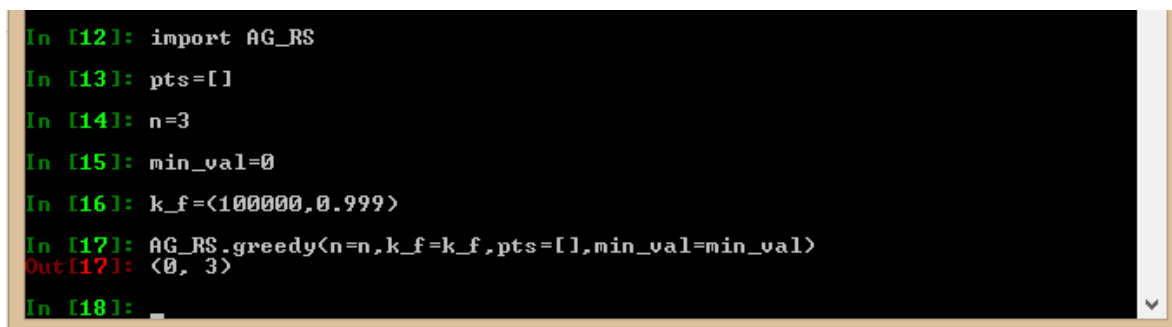
Al realizar esto aparecerá una pantalla en blanco, dar clic derecho del ratón sobre la pantalla que apareció para acercar y clic izquierdo para alejar.



Algoritmos RS y Glotón

Para la ejecución del algoritmo glotón se importa el archivo AG_RS,

- 1.-Se ingresa un arreglo de puntos vacío
- 2.- Se ingresa el número de vértices del grafo $n=3$.
- 3.-Se ingresa el número de cruces a calcular, este será la condición de paro de los algoritmos, aunque dentro del algoritmo también está el número de ejecuciones a calcular.
- 4.- Se ingresa el valor de aceptación de las nuevas configuraciones, este valor realiza la reducción del número de cruces y le da la característica de algoritmo glotón
- 5.-al finalizar regresa el número de cruces obtenido y el número de vértices del grafo que se ingresó.



Algoritmo RS

Para la ejecución del algoritmo RS se importa el archivo AG_RS.

- 1.-Se ingresa un arreglo de puntos vacío.
- 2.- Se ingresa el número de vértices del grafo $n=3$.
- 3.-Se ingresa el número de cruces a calcular, este será la condición de paro de los algoritmos, aunque dentro del algoritmo también está el número de ejecuciones a calcular.
- 4.- Se ingresa la temperatura inicial del algoritmo k_f .
- 5.-se ingresa la temperatura de enfriamiento T , esta temperatura tomara valores que aumenta el número de cruces.
- 5.-al finalizar regresa el número de cruces obtenido y el número de vértices del grafo que se ingresó.

```
In [26]: import AG_RS
In [27]: pts=[]
In [28]: n=3
In [29]: k_f=(100000,0.999)
In [30]: T=(100000,0.995)
In [31]: AG_RS.simulated_annealing(n=n,pts=pts,k_f=k_f,T=T,min_val=min_val)
Out[31]: (3, 0)
In [32]: _
```

Para los tres algoritmo se puede utilizar el visualizador comentado en el primer algoritmo.