Ejercicios3

April 22, 2018

In [1]:
```
#Ejercicio 5
```

In [25]:
```
def matar(L):
    for i in xsrange(len(L)):
        if L[i] == 0:
            continue
        x = randint(0,len(L) - 1)
        while i == x and L[i] != 0:
            x = randint(0,len(L) - 1)
        L[x] = 0
    return [1 for int in xsrange(L.count(1))]
def jugar(N):
    L = [ 1 for int in xsrange(N)]
    while(len(L) > 1):
        L = matar(L)
    return L.count(1),L
```

In [26]:
```
def comprobar_1():
    res = 0
    for int in xsrange(1000):
        r,L = jugar(100)
        if r == 1:
            res = res + 1
    return res,res/1000
```

In [27]:
```
comprobar_1()
```

Out[27]:
```
(1000, 1)
```

In [28]:
```
#Ejercicio 6
```

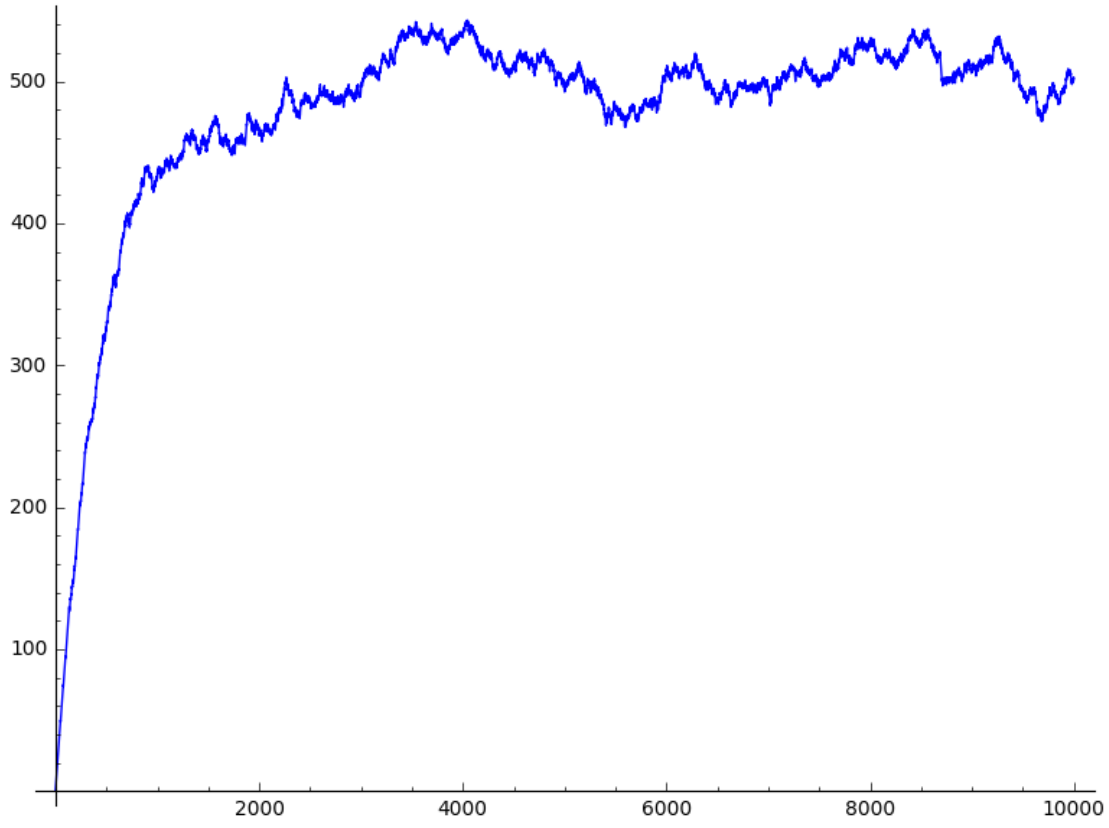
In [39]:
```
# a y b
def siguiente(n,La,Lb):
    p = randint(0,n-1)
    if La[p] == 1:
        La[p] = 0
        Lb[p] = 1
    else:
```

```
            La[p] = 1
            Lb[p] = 0
        return La,Lb
```

In [101]:
```
def perros_pulgas(n,t):
    La = [1 for int in xsrange(n)]
    Lb = [0 for int in xsrange(n)]
    L = []
    for i in xsrange(t):
        La,Lb = siguiente(n,La,Lb)
        L.append((i,Lb.count(1)))
    return L,La.count(1),Lb.count(1)
```

In [102]:
```
L,a,b = perros_pulgas(1000,10000)
line2d(L)
```

Out[102]:



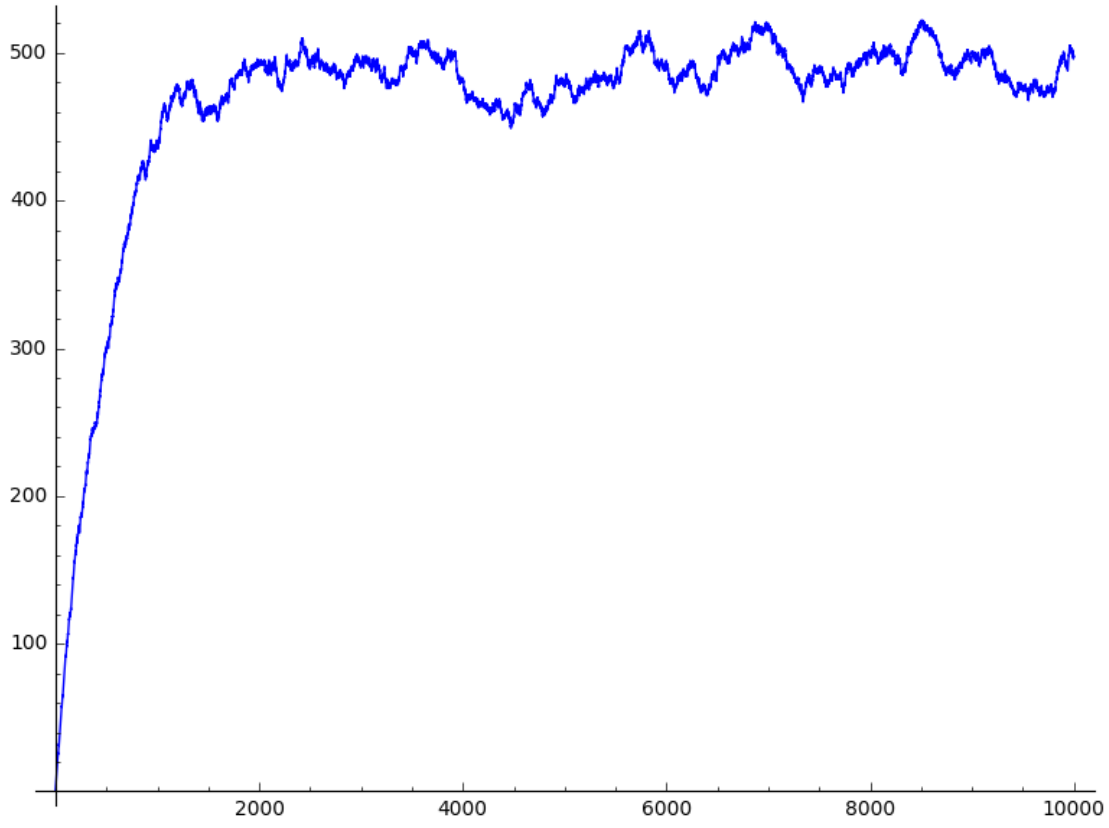
In [103]:
```
#c
def siguiente1(n,nb):
    p = randint(0,n-1)
```

```
            if p > nb:
                nb = nb + 1
            else:
                 nb = nb - 1
            return nb
```

In [104]:
```
def perros_pulgas1(n,t):
    nb = 0
    L = []
    for i in xsrange(t):
        nb = siguiente1(n,nb)
        L.append((i,nb))
    return L,nb
```

In [105]:
```
L,nb = perros_pulgas1(1000,10000)
line2d(L)
```

Out[105]:



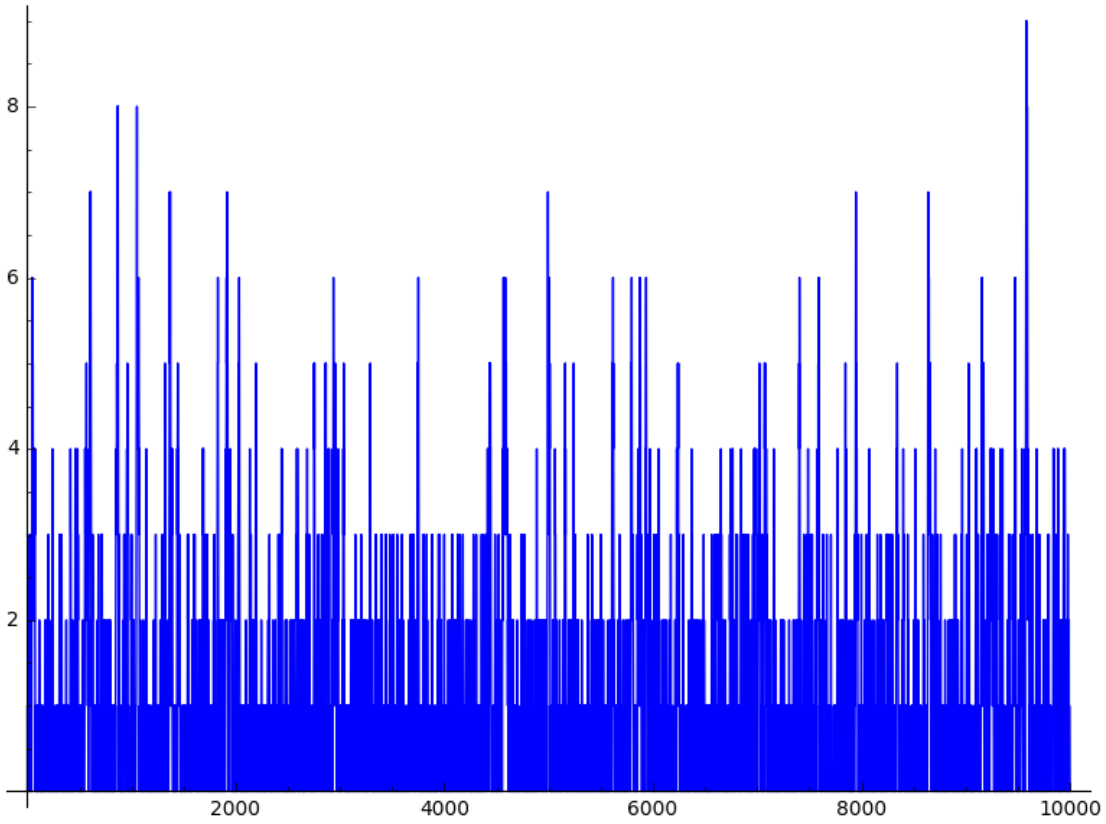
In [112]:
```
#d
def siguiente2(n,pa,pb,nb):
```

```
            perro = randint(0,1)
            if perro == 0:
                if random() < pa:
                    nb = nb + 1
            else:
                if random() < pb:
                    if nb != 0:
                        nb = nb - 1
            return nb
```

In [113]:
```
def perros_pulgas2(n,t,pa,pb):
    nb = 0
    L = []
    for i in xsrange(t):
        nb = siguiente2(n,pa,pb,nb)
        L.append((i,nb))
    return L,nb
```

In [116]:
```
L,nb = perros_pulgas2(1000,10000,0.5,1)
line2d(L)
```

Out[116]:



```
In [250]: #Ejercicio 7
          def barajar(L):
              barajar = [randint(0,1) for int in xsrange(len(L))]
              k = barajar.count(0)
              L1 = L[:k]
              L2 = L[k:]
              L3 = [0 for int in xsrange(len(L))]
              l1 = 0
              l2 = 0
              for i in xsrange (len(L)):
                  if barajar[i] == 0:
                      L3[i] = L1[l1]
                      l1 = l1 + 1
                  else:
                      L3[i] = L2[l2]
                      l2 = l2 + 1
              return L3

In [251]: def entropia(n):
              return n*log(n,base = 2)
          def Il(L):
              sum = 0
              for item in L:
                  sum = sum + entropia(item)
              return -sum.n()

In [257]: k = [1 for int in xsrange(7)]
          k[0] = 2
          for int in xsrange(100):
              k = barajar(k)
              print k
              print Il(barajar(k))

[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 1, 1, 2, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 2, 1, 1, 1]
```

```
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 2, 1, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 2, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 2, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
-2.00000000000000
[1, 1, 1, 1, 2, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
```

```
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
-2.00000000000000
[1, 1, 1, 1, 2, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 2, 1, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 2, 1, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 2, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
```

```
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 1, 2, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 2, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 2, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 2, 1, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
```

```
-2.00000000000000
[1, 1, 1, 1, 1, 1, 2]
-2.00000000000000
[1, 1, 1, 1, 2, 1, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 1, 1, 2, 1, 1]
-2.00000000000000
[1, 2, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 1, 1, 1, 2, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[1, 1, 2, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
[2, 1, 1, 1, 1, 1, 1]
-2.00000000000000
```

In [ ]: