Question 1. Count how many times out of 1000 trials the drunken man returns to origin within 100 steps. Use return1d.py and extend it to 2D and 3D. Compare the probabilities of returning to origin for different dimensions.

In [1]:

```
import random
import numpy as np
import matplotlib.pyplot as plt
```

Trial : the number of drunken men

N : the number of steps

In [2]:

```
Trial=10000
N=int(1000)
```

location array for each time

In [3]:

```
x=np.array(([0]*2)*N).reshape((N,2))
x
```

Out[3]:

```
array([[0, 0],
       [0, 0],
       [0, 0],
       ...,
       [0, 0],
       [0, 0],
       [0, 0]])
```

Set the random variables

In [4]:

```
# direction of each step
left = np.array([-1,0])
right = np.array([1,0])
up = np.array([0,1])
down = np.array([0,-1])
```

Simulation in 2D

In [7]:

```
cnt=0 # the number of drunken men who returns to origin within 100 steps
for t in range(Trial): # for each drunken man
    x = 0*x  # initialize location array
    for i in range(N-1): # trial
        r=random.choice([left, right, up, down]) # random variables
        x[i+1] = x[i] + r
#         if t == 0 and i <= 5:
#             print(r,x[i])
        if all(x[i+1] == np.zeros(2)) : # if x[i+1][0] == 0 and x[i+1][1] == 0
            #print(t,i)
            cnt +=1
            break
        if i > 100 : # we consider how many persons back to origin wthin '100' steps
            break
```

In [8]:

```
print(cnt/Trial)
```

```
0.5917
```

3D

In [9]:

```
x=np.array(([0.]*3)*N).reshape((N,3))
x
```

Out[9]:

```
array([[0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       ...,
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.]])
```

In [10]:

```
# direction of each step
forward = np.array([1.,0.,0.])
back = np.array([-1.,0.,0.])
left = np.array([0.,1.,0.])
right = np.array([0.,-1.,0.])
up = np.array([0.,0.,1.])
down = np.array([0.,0.,-1.])
```

In [11]:

```
cnt=0
for t in range(Trial):
    x = 0*x
    for i in range(N-1):
        r=random.choice([forward, back,left, right, up, down])
        x[i+1] = x[i] + r
#         if t == 0 and i <= 5:
#             print(r,x[i])
        if all(x[i+1] == np.zeros(3)) :
            #print(t,i)
            cnt +=1
            break
        if i > 100 : # we consider how many persons back to origin wthin '100' steps
            break
```

In [12]:

```
print(cnt/Trial)
```

```
0.3136
```