In [3]:

```
import numpy as np
import math
import matplotlib.pyplot as plt
import pandas as pd
```

Question 2

In [115]:

```
# Assuming prob 0.1 of a person being censored at every time instant
# considering 100 people initially and 50 deaths. Sampling from a discrete gemoetric distribution (analogous

NUM_PEOPLE_1=100
NUM_PEOPLE_2=100
NUM_DEATHS=50
RAND=5
RAND_INDICES=[0.001, 0.009, 0.02, 0.035, 0.045]
DEATHS_DICT_1={}
DEATHS_DICT_2={}
MAX_DEATH_DURATION=-100

DEATHS_GROUP_1=np.zeros(NUM_DEATHS, dtype=np.uint8)
DEATHS_GROUP_2=np.zeros(NUM_DEATHS, dtype=np.uint8)

for death in range(NUM_DEATHS):
    r_gamma=np.random.randint(RAND)
    death_period=np.random.geometric(RAND_INDICES[r_gamma])
    DEATHS_GROUP_1[death]=death_period

    r_gamma=np.random.randint(RAND)
    death_period=np.random.geometric(RAND_INDICES[r_gamma])
    DEATHS_GROUP_2[death]=death_period

DEATHS_GROUP_1=list(DEATHS_GROUP_1)
DEATHS_GROUP_2=list(DEATHS_GROUP_2)

DEATHS_GROUP_1.sort()
DEATHS_GROUP_2.sort()

for death in range(NUM_DEATHS):
    if(MAX_DEATH_DURATION<DEATHS_GROUP_1[death]):
        MAX_DEATH_DURATION=DEATHS_GROUP_1[death]

    if(DEATHS_GROUP_1[death] in DEATHS_DICT_1.keys()):
        DEATHS_DICT_1[DEATHS_GROUP_1[death]]+=1
    else:
        DEATHS_DICT_1[DEATHS_GROUP_1[death]]=1

    if(MAX_DEATH_DURATION<DEATHS_GROUP_2[death]):
        MAX_DEATH_DURATION=DEATHS_GROUP_2[death]
    if(DEATHS_GROUP_2[death] in DEATHS_DICT_2.keys()):
        DEATHS_DICT_2[DEATHS_GROUP_2[death]]+=1
    else:
        DEATHS_DICT_2[DEATHS_GROUP_2[death]]=1


INSTANT=[]
ALIVE_1=[100]
ALIVE_2=[100]
CENSORED_1=[]
CENSORED_2=[]
DEAD_1=[]
DEAD_2=[]

for time in range(MAX_DEATH_DURATION):
    INSTANT.append(time)

    COUNT_1=1
    COUNT_2=1

    if(time in DEATHS_DICT_1):
        if(NUM_PEOPLE_1-DEATHS_DICT_1[time]>=0):
            NUM_PEOPLE_1=NUM_PEOPLE_1-DEATHS_DICT_1[time]

    if(time in DEATHS_DICT_2):
        if(NUM_PEOPLE_2-DEATHS_DICT_2[time]>=0):
            NUM_PEOPLE_2=NUM_PEOPLE_2-DEATHS_DICT_2[time]

    for one in range(NUM_PEOPLE_1):
        if(0.1>np.random.uniform()):
            COUNT_1=COUNT_1+1
            if(NUM_PEOPLE_1>=1):
                NUM_PEOPLE_1=NUM_PEOPLE_1-1

    for one in range(NUM_PEOPLE_2):
        if(0.1>np.random.uniform()):
            COUNT_2=COUNT_2+1
            if(NUM_PEOPLE_2>=1):
                NUM_PEOPLE_2=NUM_PEOPLE_2-1

    ALIVE_1.append(num_alive_1)
    ALIVE_2.append(num_alive_2)

    if(time in DEATHS_DICT_1):
        DEAD_1.append(DEATHS_DICT_1[time])
        W_1=DEATHS_DICT_1[time]
    else:
        DEAD_1.append(0)
        W_1=0

    if(time in DEATHS_DICT_2):
        DEAD_2.append(DEATHS_DICT_2[time])
        W_2=DEATHS_DICT_2[time]
    else:
        DEAD_2.append(0)
        W_2=0

    CENSORED_1.append(COUNT_1)
    CENSORED_2.append(COUNT_2)

    if(NUM_PEOPLE_2+NUM_PEOPLE_1+COUNT_1+COUNT_2+W_2+W_1==0):
        break

data_array=np.asarray([INSTANT, ALIVE_1[:len(ALIVE_1)], DEAD_1, CENSORED_1, ALIVE_2[:len(ALIVE_2)], DEAD_2,
```

In [128]:
```
data_chart=pd.DataFrame(data_array.transpose(), columns=['Time', 'Group 1 Alive', "Group 1 Died", "Group 1 C
data_chart
```

Out[128]:

	Time	Group 1 Alive	Group 1 Died	Group 1 Censored	Group 2 Alive	Group 2 Died	Group 2 Censored
0	0	100	0	13	100	0	10
1	1	87	0	6	90	1	5
2	2	81	0	13	84	0	13
3	3	68	2	10	71	3	5
4	4	56	0	6	63	1	8
5	5	50	2	6	54	1	4
6	6	42	0	1	49	1	4
7	7	41	1	9	44	0	5
8	8	31	0	0	39	3	2
9	9	31	2	3	34	1	5
10	10	26	1	1	28	2	1
11	11	24	0	3	25	0	2
12	12	21	0	1	23	0	2
13	13	20	0	3	21	1	1
14	14	17	0	1	19	0	2
15	15	16	0	2	17	1	0
16	16	14	0	2	16	2	1
17	17	12	0	2	13	0	3
18	18	10	0	1	10	1	0
19	19	9	1	1	9	1	1
20	20	7	1	0	7	1	0
21	21	6	0	0	6	1	0
22	22	6	0	0	5	0	0

	Time	Group 1 Alive	Group 1 Died	Group 1 Censored	Group 2 Alive	Group 2 Died	Group 2 Censored
23	23	6	1	1	5	2	0
24	24	4	0	0	3	1	0
25	25	4	2	0	2	0	0
26	26	2	0	1	2	0	0
27	27	1	1	0	2	0	1
28	28	0	1	0	1	0	0
29	29	0	1	0	1	0	0
30	30	0	0	0	1	0	0
31	31	0	0	0	1	1	0
32	32	0	0	0	0	0	0

```
In [ ]:
```