MA326 Lab 1

Ashish Kumar Barnawal
180123006

September 9, 2020

Question 1 Generate the sequence of numbers x_i for $a = 6, b = 0, m = 11$, and x_0 ranging from 0 to 10. Also, generate the sequence of numbers x_i for $a = 3, b = 0, m = 11$, and x_0 ranging from 0 to 10. Observe the sequence of numbers generated and observe the repetition of values. Tabulate these for each group of values. How many distinct values appear before repetitions? Which, in your opinion, are the best choices and why?

Observation For $a = 6$, $b = 0$, $m = 11$, 10 distinct values appear
For $a = 3$, $b = 0$, $m = 11$, 5 distinct values appear

The first case is a better choice because

- It takes more terms before repetition occurs, i.e. the period is longer
- There are more distinct values
- The sub-intervals also have similar or better uniform distribution in the first case than the second

Table 1: $a = 6$

x_0	sequence x_i													
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	6	3	7	9	10	5	8	4	2	1	6	3	7
2	2	1	6	3	7	9	10	5	8	4	2	1	6	3
3	3	7	9	10	5	8	4	2	1	6	3	7	9	10
4	4	2	1	6	3	7	9	10	5	8	4	2	1	6
5	5	8	4	2	1	6	3	7	9	10	5	8	4	2
6	6	3	7	9	10	5	8	4	2	1	6	3	7	9
7	7	9	10	5	8	4	2	1	6	3	7	9	10	5
8	8	4	2	1	6	3	7	9	10	5	8	4	2	1
9	9	10	5	8	4	2	1	6	3	7	9	10	5	8
10	10	5	8	4	2	1	6	3	7	9	10	5	8	4

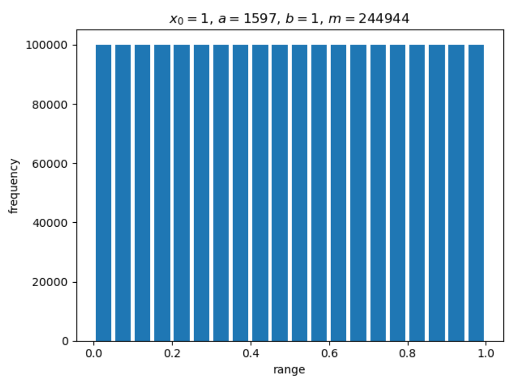
Table 2: $a = 3$

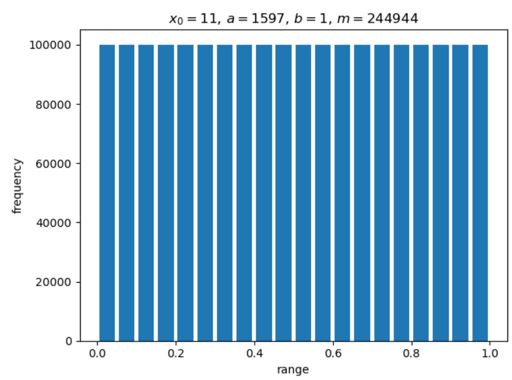
x_0	sequence x_i													
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	3	9	5	4	1	3	9	5	4	1	3	9	5
2	2	6	7	10	8	2	6	7	10	8	2	6	7	10
3	3	9	5	4	1	3	9	5	4	1	3	9	5	4
4	4	1	3	9	5	4	1	3	9	5	4	1	3	9
5	5	4	1	3	9	5	4	1	3	9	5	4	1	3
6	6	7	10	8	2	6	7	10	8	2	6	7	10	8
7	7	10	8	2	6	7	10	8	2	6	7	10	8	2
8	8	2	6	7	10	8	2	6	7	10	8	2	6	7
9	9	5	4	1	3	9	5	4	1	3	9	5	4	1
10	10	8	2	6	7	10	8	2	6	7	10	8	2	6

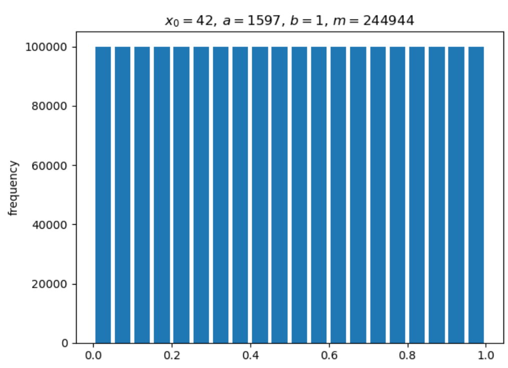
Question 2 Generate a sequence u_i with $m = 244944, a = 1597, 51749$ (choosing x_0 as per your choice). Then group the values in the ranges $0 - 0.05, 0.05 - 0.10, 0.10 - 0.15...$ and observe their frequencies (i.e., the number of values falling in each group). For 5 different x_0 values, tabulate the frequencies in each case, draw the bar diagrams for these data and put in your observations

---

Observation For given m and $b = 1$, $a = 1597$ satisfies the Knuth condition, while $a = 51749$ doesn't. From the graphs, it can be seen that the graphs of $a = 1597$ are flat while for $a = 51749$ it is not always so.







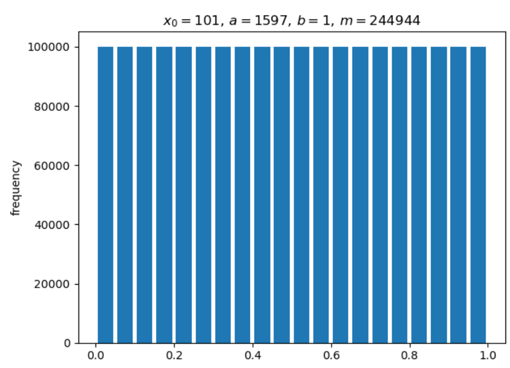
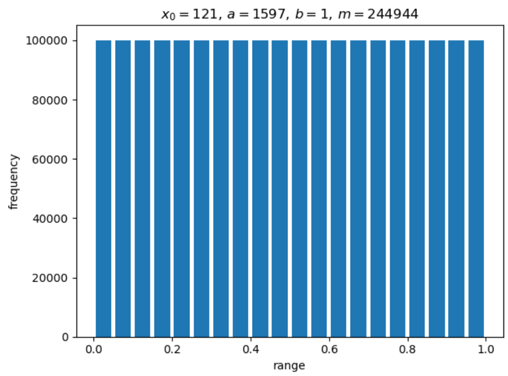
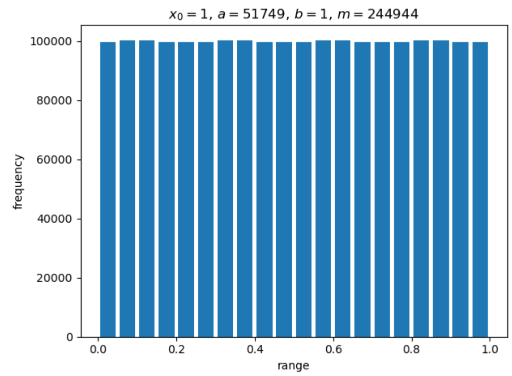
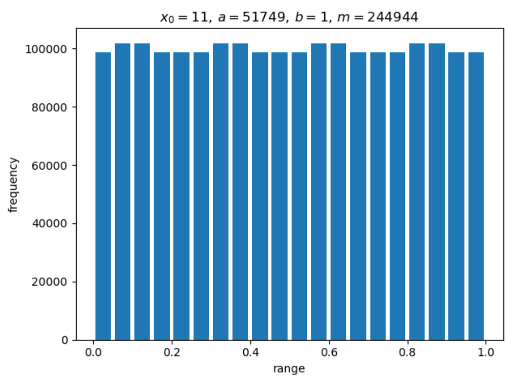


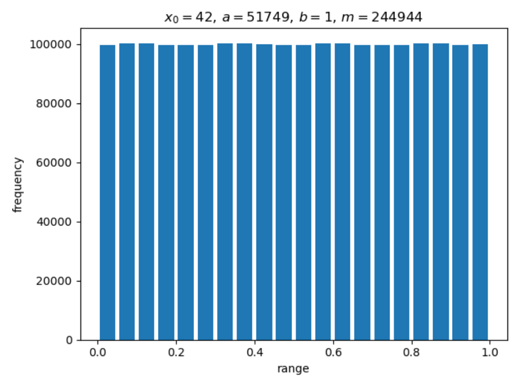
Table 3: Frequency table for $a = 1597$

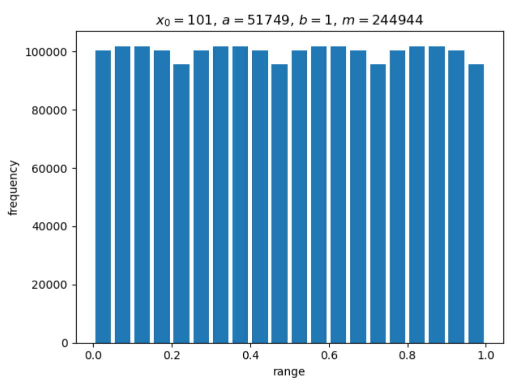
	x_0				
	1	11	42	101	121
0.00-0.05	99973	99918	99989	100037	99929
0.05-0.10	99933	99989	100045	99931	99956
0.10-0.15	100023	100035	100045	99948	100045
0.15-0.20	100020	100036	100011	100031	100025
0.20-0.25	100031	100006	99936	100057	100050
0.25-0.30	99987	99927	99997	100036	99927
0.30-0.35	99907	100002	100027	99933	99958
0.35-0.40	100022	100031	100038	99955	100012
0.40-0.45	100024	100049	100001	100040	100059
0.45-0.50	100050	100005	99909	100035	100035
0.50-0.55	99979	99934	99992	100048	99942
0.55-0.60	99921	99986	100040	99922	99957
0.60-0.65	100030	100054	100032	99957	100049
0.65-0.70	100045	100025	100003	100028	100037
0.70-0.75	100040	100022	99932	100042	100035
0.75-0.80	99963	99940	99989	100042	99939
0.80-0.85	99940	99981	100037	99917	99950
0.85-0.90	100019	100022	100034	99968	100026
0.90-0.95	100049	100037	100033	100047	100031
0.95-1.00	100044	100001	99910	100026	100038


$x_0 = 121$, $a = 1597$, $b = 1$, $m = 244944$
frequency
range


$x_0 = 1$, $a = 51749$, $b = 1$, $m = 244944$
frequency
range


$x_0 = 11$, $a = 51749$, $b = 1$, $m = 244944$
frequency
range


$x_0 = 42$, $a = 51749$, $b = 1$, $m = 244944$
frequency
range


$x_0 = 101$, $a = 51749$, $b = 1$, $m = 244944$
frequency
range

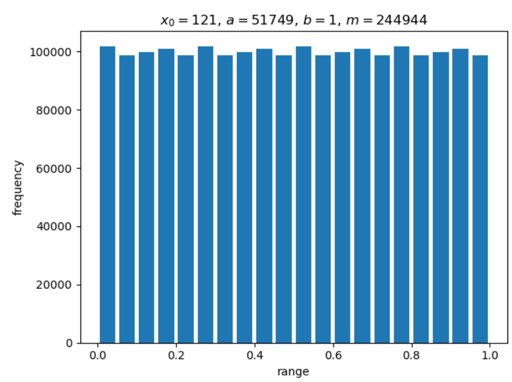
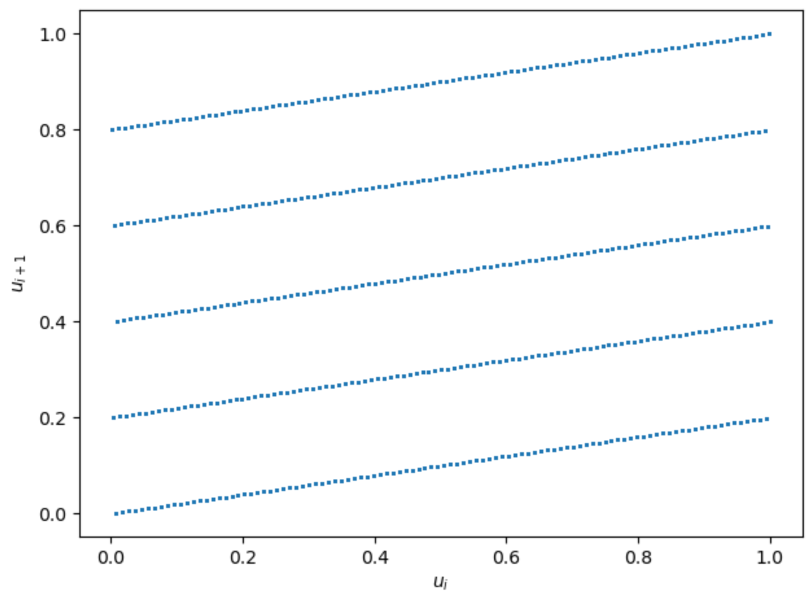

$x_0 = 121$, $a = 51749$, $b = 1$, $m = 244944$
frequency
range

Table 4: Frequency table for $a = 51749$

	x_0				
	1	11	42	101	121
0.00-0.05	99785	98766	99783	100308	101865
0.05-0.10	100316	101855	100307	101851	98752
0.10-0.15	100302	101856	100312	101852	99797
0.15-0.20	99796	98769	99789	100305	100828
0.20-0.25	99797	98761	99795	95681	98756
0.25-0.30	99796	98764	99801	100306	101861
0.30-0.35	100301	101851	100302	101854	98762
0.35-0.40	100310	101850	100306	101852	99794
0.40-0.45	99785	98764	99808	100311	100826
0.45-0.50	99793	98765	99777	95677	98773
0.50-0.55	99795	98768	99804	100314	101834
0.55-0.60	100307	101855	100306	101847	98781
0.60-0.65	100311	101856	100306	101855	99787
0.65-0.70	99796	98769	99796	100308	100822
0.70-0.75	99792	98764	99795	95681	98776
0.75-0.80	99801	98764	99789	100308	101840
0.80-0.85	100309	101846	100316	101855	98769
0.85-0.90	100307	101848	100311	101848	99800
0.90-0.95	99799	98763	99786	100310	100817
0.95-1.00	99802	98766	99811	95677	98760

Question 3 Generate a sequence u_i with $a = 1229, b = 1, m = 2048$. Plot in a two-dimensional graph the points (u_{i-1}, u_i), i.e., the points $(u_1, u_2), (u_2, u_3), (u_3, u_4), \ldots$

Observation $a = 1229$, $m = 2048$, $b = 1$ satifies the Knuth condition. On plotting we get



Programs

Setup Instructions

Install `python3`, and `matplotlib` library for graphs. Create a folder named `output` in the same directory as the python files. The program will output all the csv files and the graphs in that directory.

Question 1

```
import sys
import os


def genseq(a, b, m, seed=1, num=20):
        x = seed % m
        res = []
        for _ in range(num):
                        res.append(x)
                        x = (a*x + b)%m
        return res

seq_size = 14

a = 6; b = 0; m = 11
res1 = []
for x0 in range(m):
        seq = genseq(a, b, m, x0, seq_size)
        res1.append(seq)

with open(os.path.join('output', 'q1o1.csv'), 'w') as
   fp:
        output = "$x0$\n"

        for x0 in range(m):
                output += f'{x0},'
                output += ','.join(list(map(str, res1[
                   x0])))
```

```
            output += '\n'
        fp.write(output)

a=3
res2 = []
for x0 in range(m):
        seq = genseq(a, b, m, x0, seq_size)
        res2.append(seq)

with open(os.path.join('output', 'q1o2.csv'), 'w') as
    fp:
        output = "$x0$\n"

        for x0 in range(m):
            output += f'{x0},'
            output += ','.join(list(map(str, res2[
                x0])))
            output += '\n'
        fp.write(output)
```

Question 2

```
from matplotlib import pyplot as plt
import sys
import os

def genseq(a, b, m, seed=1, num=20):
        x = seed % m
        res = []
        for _ in range(num):
                res.append(x/float(m))
                x = (a*x + b)%m
        return res

sample_size = int(2e6)
m = 244944
b = 1


# 5 different x0, a=1597, 51749
ind = 0
txtind = 0
x0vals = [1, 11, 42, 101, 121]
avals = [1597, 51749]

for a in avals:
        res = []
        for x0 in x0vals:
                ind += 1
                seq = genseq(a, b, m, x0, sample_size)
                hist_data, bin_edges, _ = plt.hist(seq,
                    range=[0, 1], bins=int(1/0.05),
                   rwidth=0.8)
                res.append(hist_data)
                plt.xlabel('range')
                plt.ylabel('frequency')
                plt.title(f'$x_0␣=␣{x0},\\/a␣=␣{a},\\/b
                   ={b},\\/m={m}$')
```

```
            plt.tight_layout()
            plt.savefig(os.path.join('output', f'
               q2o{ind}.png'))
            plt.clf()

    with open(os.path.join('output', f'q2a{a}.csv')
       , 'w') as fp:
            fp.write("$x_0$,")
            fp.write(','.join(list(map(lambda x:
               str(int(x)), x0vals))))
            fp.write('\n')

            for itr in range(len(bin_edges) - 1):
                    fp.write(f"{bin_edges[itr]:.2f
                       }-{bin_edges[itr+1]:.2f},")
                    row = ','.join([str(int(res[i][
                       itr])) for i in range(len(
                       x0vals))])
                    fp.write(row)
                    fp.write('\n')
```

Question 3

```
import matplotlib.pyplot as plt
import os

def genseq(a, b, m, seed=1, num=20):
    x = seed % m
    res = []
    for _ in range(num):
        x = (a*x + b)%m
        res.append(x/float(m))
    return res

a = 1229
b = 0
m = 2048
x0 = 123
numpts = 4196

seq = genseq(a, b, m, x0, numpts+1)

plt.scatter(seq[:-1], seq[1:], 1)
plt.xlabel('$u_i$')
plt.ylabel('$u_{i+1}$')
plt.tight_layout()
plt.savefig(os.path.join('output', 'q3out0.png'))
plt.show()
```