ALGO project Test Case results

Simple case no reappearance dependency:

g1 d1 case:

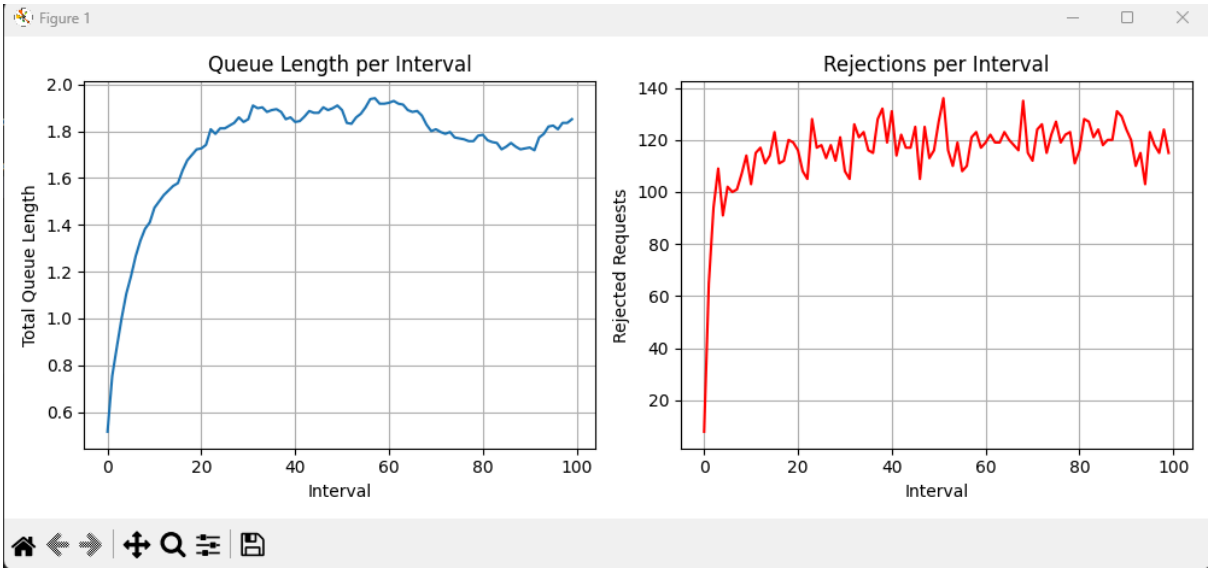
G1 and d4 case to show how greedy approach better for reappearance dependency:

Large dataset:

Refer to Test Case 3 for INput Params
d= 2 and g = 4

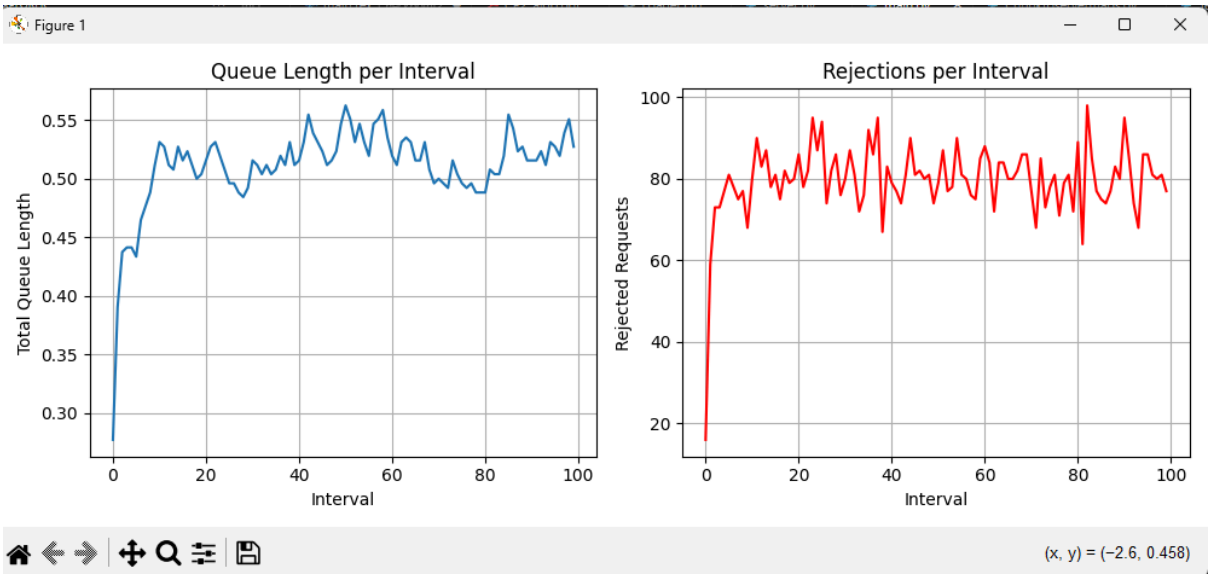
Random:

g= 1 case



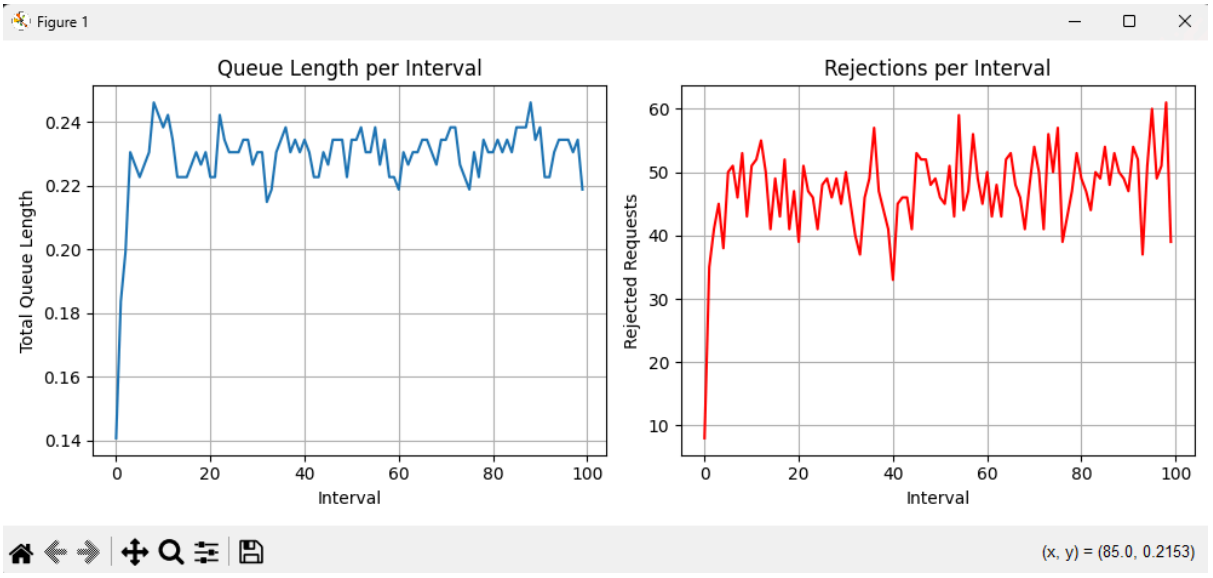
```
--- Simulation Summary ---
Total Requests: 25600
Accepted Requests: 14035
Rejected Requests: 11565
Rejection Rate: 0.4518
```

G = 2 case




```
--- Simulation Summary ---
Total Requests: 25600
Accepted Requests: 17636
Rejected Requests: 7964
Rejection Rate: 0.3111
```

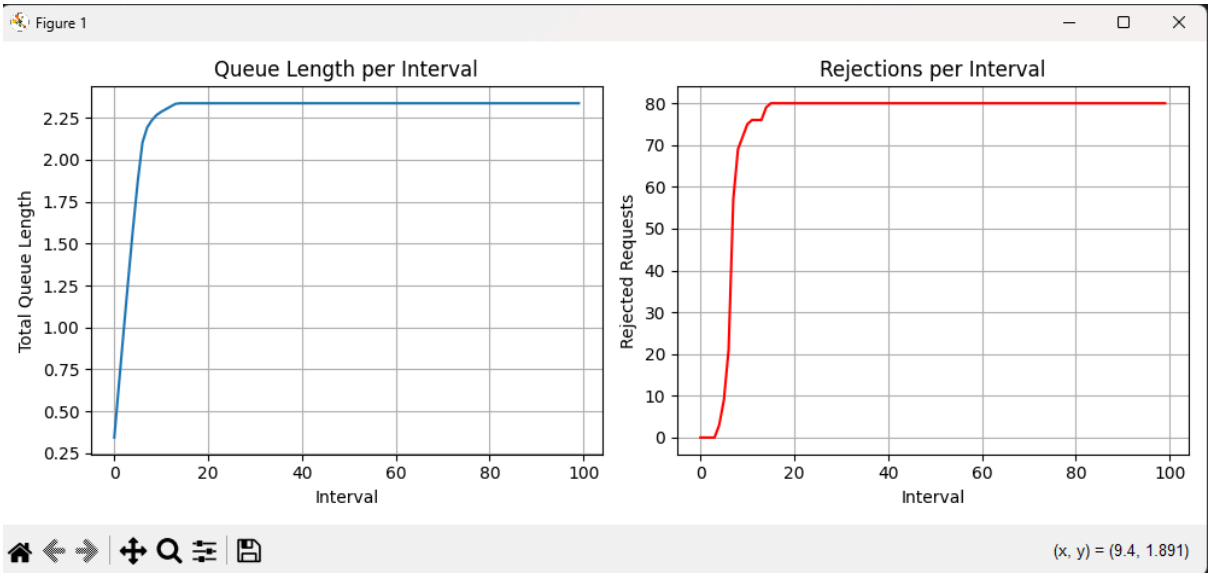
G =4



```
--- Simulation Summary ---
Total Requests: 25600
Accepted Requests: 20886
Rejected Requests: 4714
Rejection Rate: 0.1841
```

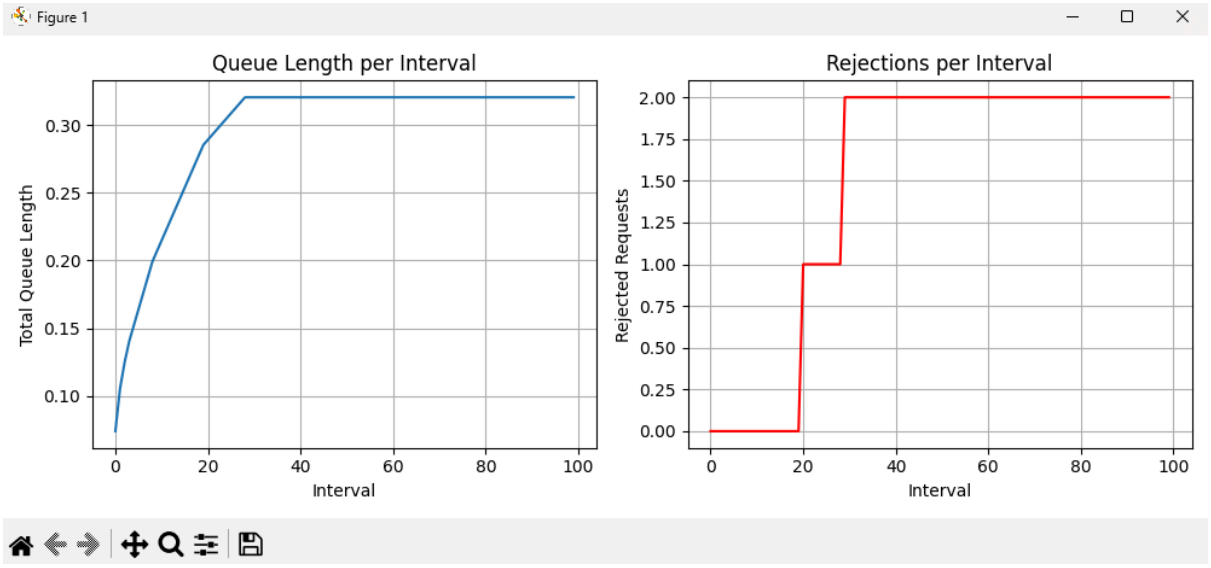
Greedy :

G =1 case



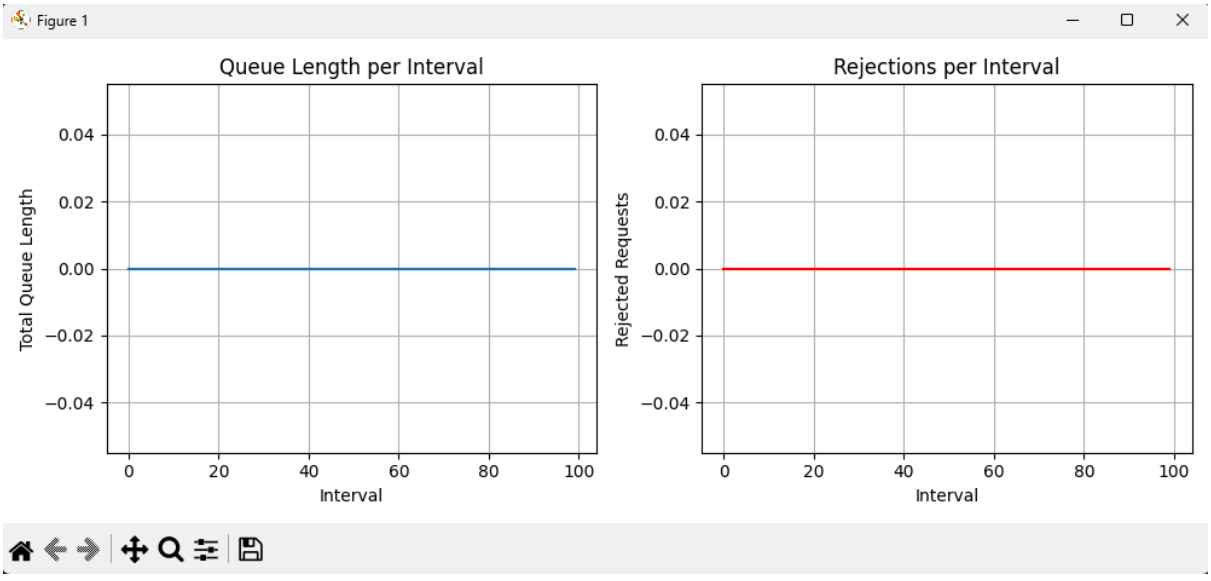

```
--- Simulation Summary ---
Total Requests: 25600
Accepted Requests: 18187
Rejected Requests: 7413
Rejection Rate: 0.2896
```

G = 2 case:




```
--- Simulation Summary ---
Total Requests: 25600
Accepted Requests: 25449
Rejected Requests: 151
Rejection Rate: 0.0059
```

G=4:

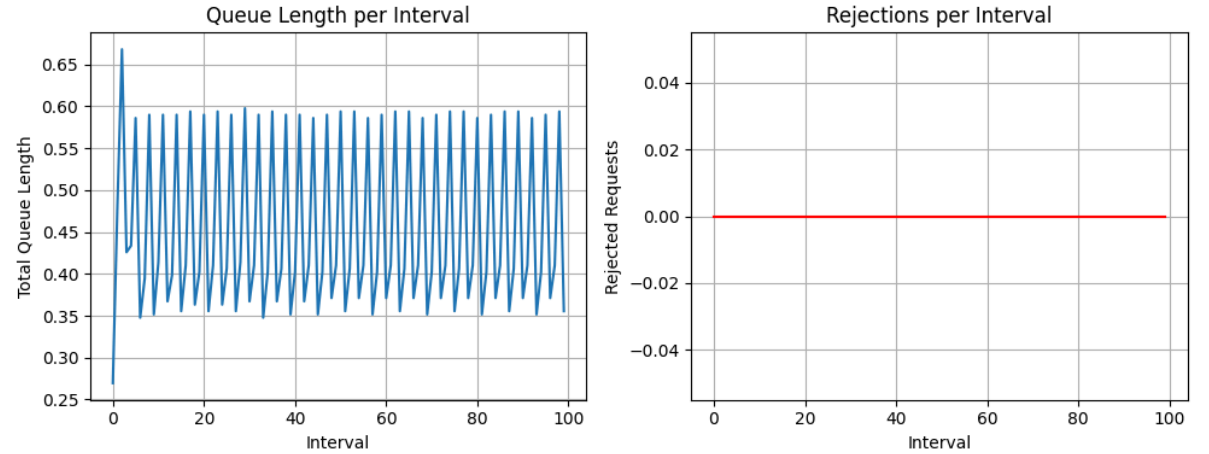



```
--- Simulation Summary ---
Total Requests: 25600
Accepted Requests: 25600
Rejected Requests: 0
Rejection Rate: 0.0000
PS D:\Algo_Project>
```

Cuckoo:

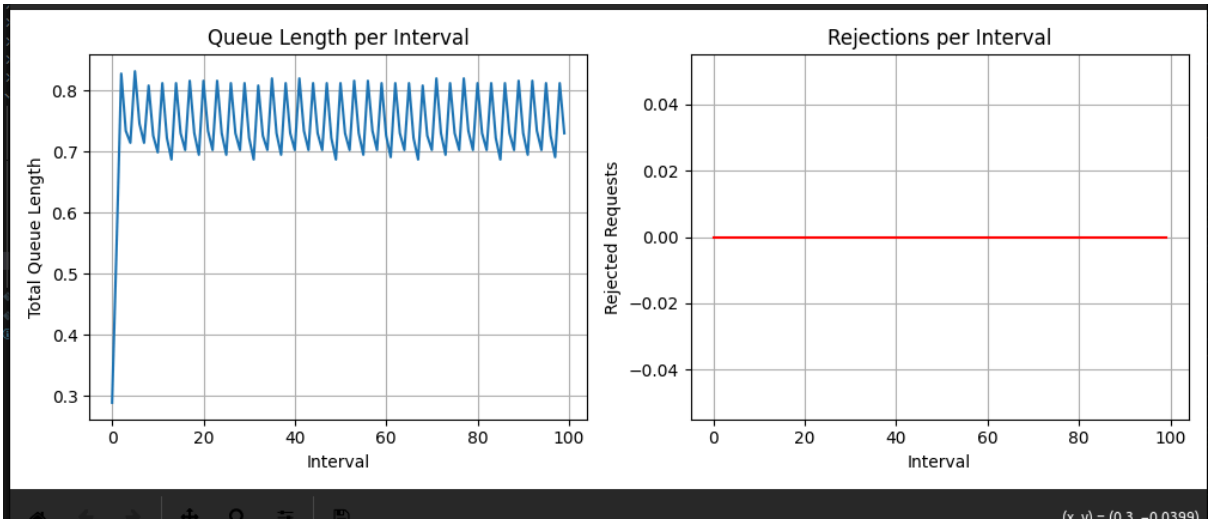
G = 4: // cuckoo hashing WOrks mainly for d = 2 and G=4 as each queue has g/4 processing (refer to research paper)

With lighter adversary reapp dependency:




```
--- Simulation Summary ---
Total Requests: 25600
Accepted Requests: 25600
Rejected Requests: 0
Rejection Rate: 0.0000
```

With severe dependency:




```
--- Simulation Summary ---
Total Requests: 25600
Accepted Requests: 25600
Rejected Requests: 0
Rejection Rate: 0.0000
```