Fadi Botros
T02

CPSC 441 – Assignment 4

About my program

My program simply takes the desired N value as a parameter and it outputs simulation stats using N budgies. My program simulates 10080 minutes (1 week) of budgies singing. It simulates the budgies using a priority queue of events. An event can either be a SONG_START or SONG_END event. Each event holds a double corresponding to when the event will occur; the time is a randomly chosen using an exponentially distributed random number generator. The events are ordered in the queue by the time that they will occur. The simulation starts off by queuing a SONG_START event for each budgie. The simulation then dequeues the events and calculates stats. If it dequeues a SONG_START event, it then queues a SONG_END event. If it dequeues a SONG_END event, it then queues a SONG_START event. Each time an event is dequeued, simulation stats are calculated. The simulation keeps going until it dequeues an event that surpasses the 10080 minutes limit.

To answer the following questions I had to introduce new functions in my program to quickly run simulations and print out stats for 1-25 budgies. I also noticed that the results from the simulation fluctuated a lot so I added a new function that runs the same simulation 10 times and averages the results. This helped make the results more consistent.

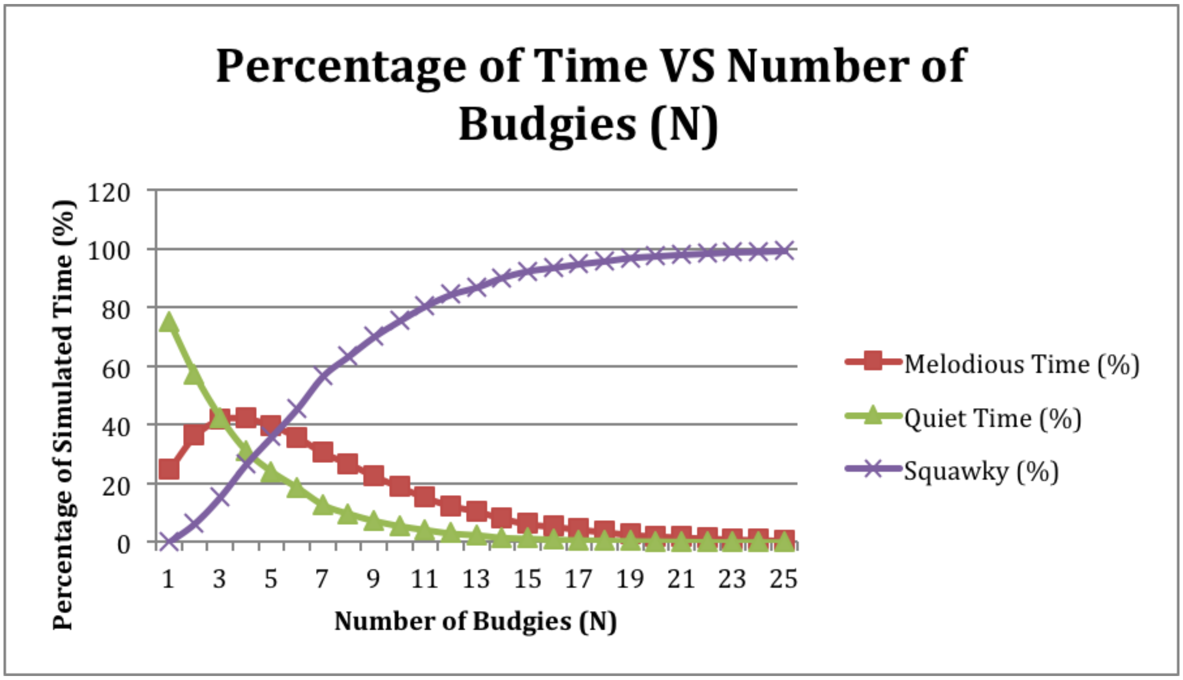
This is a sample output from my program:

```
Number of budgies: 4
Mean song length: 10.000000 minutes
Mean quiet time length: 30.000000 minutes
S Value: 0.250000
Simulated run time: 10080.000000 minutes

Melodious time:  4217.626714     (41.841535%)
Quiet time:      3305.374598     (32.791415%)
Squawky time:    2556.998689     (25.367050%)

Number of total songs: 1004.000000
Number of perfect songs: 227.000000
Percentage of perfect songs: 22.609562%
```

Question 1

For this question, I ran my simulation for 1 – 25 budgies and recorded the percentage of melodious, quiet and squawky time for each simulation. The table with all the results is shown below. I then graphed all this data; the graph is shown below as well. We can see as the number of budgies increases, the percentage of quiet time and melodious time near zero percent, while the percentage of squawky time nears 100 %. This makes sense, as the more budgies we have, the noisier it will be. We can also see that, the maximum number of melodious time is 42.194 %, which occurred when there were 4 budgies. This optimal number of budgies makes sense since if on average every budgie sings for 10 minutes and is quiet for 30 minutes, during the quiet time of one budgie, 3 other budgies could be singing. The table and graph below show that the optimal number of budgies is 4.

N	Melodious Time (%)	Quiet Time (%)	Squawky Time (%)
1	24.853051	75.146949	0
2	36.513934	57.028612	6.457454
3	41.958744	42.463577	15.57768
4	42.193818	31.196854	26.609328
5	39.893035	24.101712	36.005254
6	35.791236	18.619242	45.589522
7	30.574953	12.801331	56.623716
8	26.855747	9.782311	63.361942
9	22.57999	7.394671	70.025339
10	18.902378	5.625063	75.472559
11	15.248954	4.29764	80.453405
12	12.49809	3.120637	84.381273
13	10.630644	2.529215	86.840141
14	8.189854	1.658739	90.151407
15	6.317029	1.360785	92.322186
16	5.441677	1.039036	93.519286
17	4.460068	0.816651	94.723281
18	3.577811	0.68337	95.738819
19	2.668145	0.482385	96.84947
20	2.190588	0.392192	97.417219
21	1.764785	0.294365	97.940849
22	1.457266	0.195858	98.346875
23	1.089532	0.118153	98.792316
24	0.893915	0.149955	98.95613
25	0.653468	0.098073	99.248459

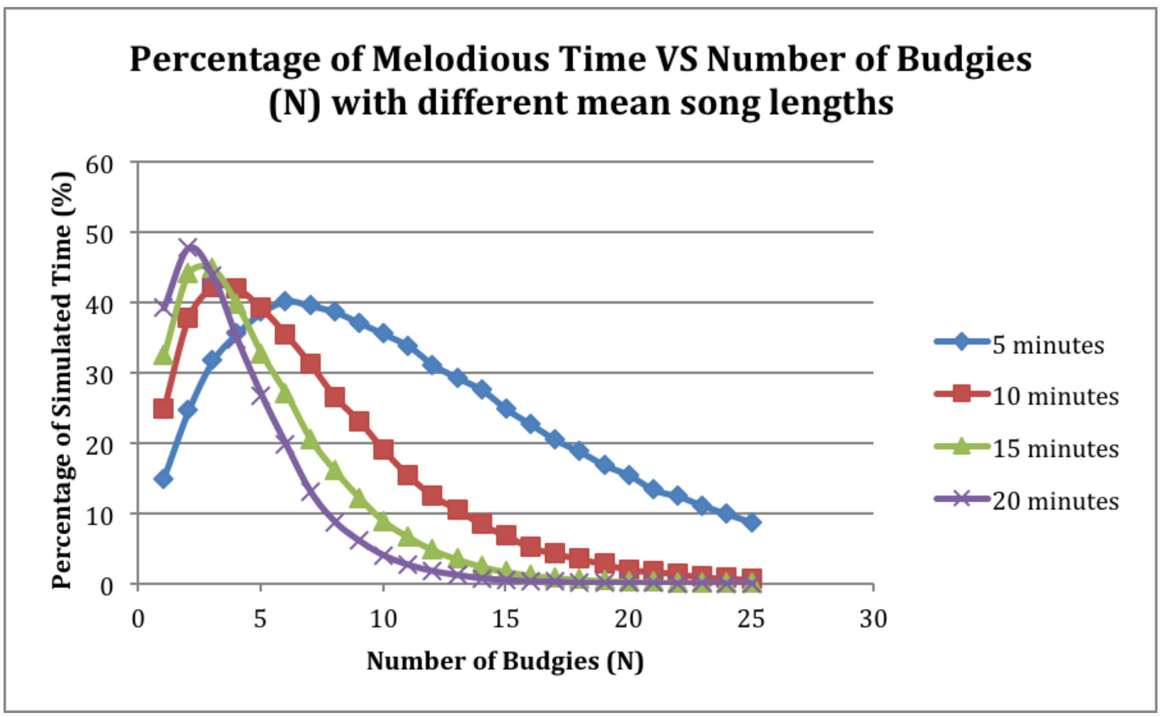


Question 2

For this question, I changed the mean song time to 5 minutes and ran the simulation for 1-25 budgies. I recorded the percentage of melodious time and repeated the process for a mean song time of 10, 15 and 20 minutes. All my results are in the table below, I also graphed the results. As we can see in the results, as the mean song time increases, the optimal of N decreases and the maximum percentage of melodious time increases.

	Percentage (%) of Melodious Time with different Mean Song Lengths			
N	5 minutes	10 minutes	15 minutes	20 minutes
1	14.958059	24.855707	32.459009	39.170811
2	24.608939	37.750684	44.124032	47.729399
3	31.735797	42.247394	44.83912	43.760453
4	35.682911	41.997653	39.740023	34.726034
5	38.638562	39.205616	32.742184	26.763723
6	40.251306	35.456223	27.074788	19.708245
7	39.608202	31.22725	20.580456	13.024341
8	38.706515	26.583825	16.070896	8.755786
9	37.152755	23.101301	12.198152	6.056854
10	35.539911	18.968799	8.861506	3.987641
11	33.841936	15.341227	6.620749	2.67577

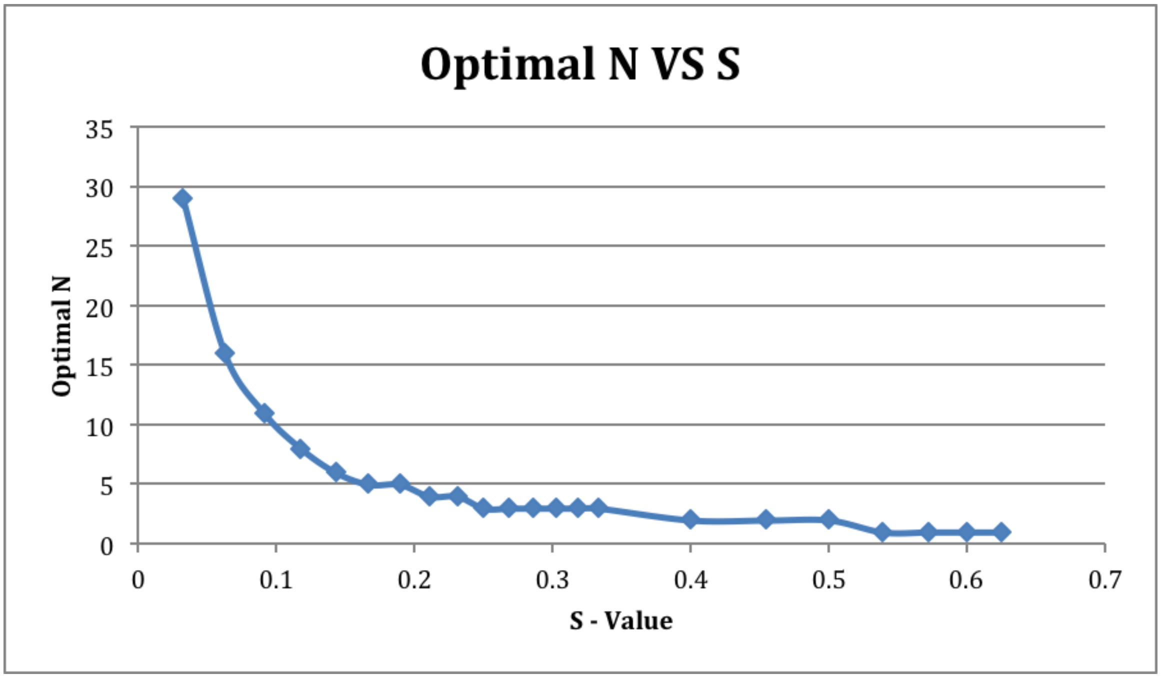
12	31.153132	12.518271	4.785903	1.791002
13	29.331483	10.560025	3.500014	1.238426
14	27.576473	8.426798	2.439315	0.759416
15	24.849018	6.805259	1.697863	0.544637
16	22.674724	5.164358	1.200108	0.341562
17	20.49168	4.307841	0.855368	0.283143
18	18.796877	3.520037	0.635119	0.118344
19	16.89284	2.843497	0.469475	0.088893
20	15.456727	2.010474	0.334958	0.106135
21	13.391758	1.764062	0.233845	0.042432
22	12.489602	1.383529	0.176082	0.024771
23	11.134352	1.000191	0.089238	0.044365
24	9.958812	0.778515	0.078239	0.021089
25	8.771193	0.622893	0.092798	0.022103



To get a better understanding of the relationship between optimal N and S, I used another technique. I started by setting the mean song length time to 1 minute and I ran the simulation for 1-25 budgies to determine the optimal number of N. I then kept increasing the mean song length by 1 and repeated the process. The table below shows all my results. I then graphed how optimal N changes with respect to S (fraction of time that a budgie sings). The graph is also shown below. As we can see, there seems to be a reciprocal relationship between S and N; **N = floor(1/S).** Floor() is there because we can't have a fraction of a budgie. Anything higher than S = 0.5, optimal N is 1.

Fadi Botros
T02

Mean Song Time (minutes)	S	Optimal N	Percentage of Melodious Time (%)
1	0.032258	29	37.687498
2	0.0625	16	37.911816
3	0.090909	11	38.830684
4	0.117647	8	39.759218
5	0.142857	6	39.47315
6	0.166667	5	40.842469
7	0.189189	5	41.273919
8	0.210526	4	41.394347
9	0.230769	4	41.10238
10	0.25	3	42.198211
11	0.268293	3	43.420734
12	0.285714	3	43.966546
13	0.302326	3	43.805237
14	0.318182	3	43.952179
15	0.333333	3	44.306357
20	0.4	2	48.509339
25	0.454545	2	49.635164
30	0.5	2	50.257413
35	0.538462	1	53.62393
40	0.571429	1	58.126015
45	0.6	1	60.059408
50	0.625	1	61.878976

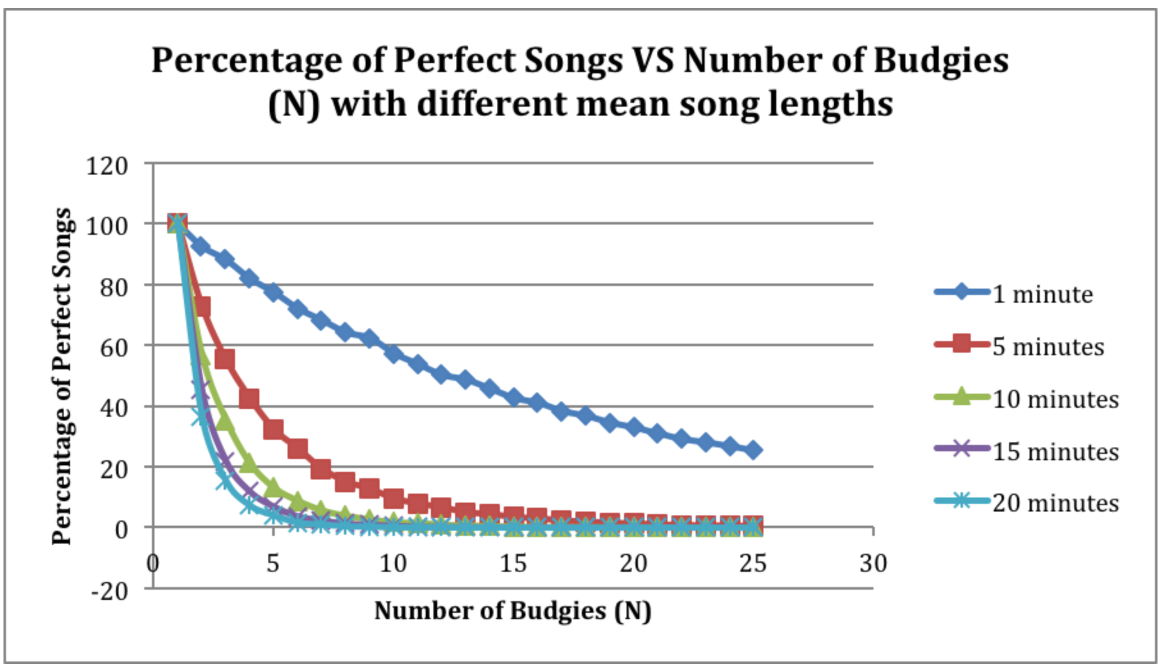


Question 3

For this question, I changed the mean song length time to 1 minute and ran the simulation for 1-25 budgies. I recorded the percentage of perfect songs and repeated the process for a mean song length time of 5, 10, 15 and 20 minutes. All my results are shown in the table below. I also graphed my results as seen below. As we can see, the percentage of perfect songs drops off quickly as the mean song length time goes up and as the number of budgies increase. This makes sense since when the mean song length time increases; budgies will start interrupting each other more frequently.
This simulation can be related to the ALOHA protocol, where the stations are equivalent to budgies, and message transmission time is equivalent to the mean song length time. The more stations we have on the network and the longer the message transmission times are, the busier the channel will be and this will lead to more collisions between transmissions.

	Percentage (%) of Perfect Songs with different Mean Song Lengths				
N	**1 minute**	**5 minutes**	**10 minutes**	**15 minutes**	**20 minutes**
1	100	100	100	100	100
2	92.52787	72.91557	56.930014	45.23702	36.654982
3	88.151802	55.273189	35.375784	21.894293	15.423535
4	81.969121	42.560858	21.124243	11.797497	7.238335
5	77.429351	32.364597	13.108525	6.831017	4.020351
6	72.111412	25.998594	8.757867	3.637845	1.574803
7	68.239077	19.36432	5.65461	2.338245	1.016853

8	64.260401	15.163704	3.822262	1.260197	0.436491
9	62.340773	12.768012	2.720219	0.929285	0.309221
10	57.347132	9.62076	1.68177	0.534879	0.074074
11	53.789113	7.919565	1.488654	0.235199	0.080866
12	50.280814	6.681573	0.891343	0.171732	0.091143
13	48.682428	5.037214	0.718391	0.124542	0.015192
14	45.714159	4.473984	0.452656	0.08602	0.014161
15	42.647931	3.463942	0.353348	0.041629	0.019962
16	41.055174	3.028849	0.175617	0.036323	0
17	38.368668	2.344881	0.153264	0.007847	0.005806
18	36.767673	1.93788	0.139288	0.017325	0.005493
19	34.5588	1.586082	0.083917	0.00942	0
20	32.930992	1.333868	0.077815	0.004476	0.004978
21	30.965408	1.033185	0.062418	0.002118	0.009446
22	29.139434	0.826706	0.019947	0.00202	0
23	28.026981	0.683306	0.025957	0.005838	0.002144
24	26.75325	0.645348	0.018131	0.003729	0
25	25.367833	0.462096	0.015987	0	0



Fadi Botros
T02

Bonus Question

For this question, I simply changed one line of code so that SONG_END events always occur after 10 minutes. I ran the simulation for 1-25 budgies and the results that I got look exactly like the results that I got in question 1. The number of optimal N is still 4. The results and the graph can be found below.
I was expecting to have the same results since the quiet time is still random and therefore songs can still start at any time. If we really wanted to increase melodious time and optimal N, we would have to control both song length and quiet time length to be fixed. With fixed song length and quiet time length, we can make it so budgies perfectly alternate singing between each other so that we have 100% melodious time.
This is similar to slotted ALOHA, where stations can only send at the beginning of equally spaced timeslots. This does indeed increases ALOHA's maximum throughput.

N	Melodious Time (%)	Quiet Time (%)	Squawky Time (%)
1	25.165488	74.834512	0
2	37.115644	56.621693	6.262663
3	41.95975	42.531074	15.509175
4	42.288654	31.406469	26.304877
5	39.448952	23.660977	36.890071
6	35.767097	17.673932	46.558971
7	31.190429	13.288314	55.521257
8	27.06037	10.281163	62.658467
9	22.738963	7.448709	69.812328
10	19.020144	5.673532	75.306324
11	15.39998	4.171143	80.428877
12	12.644572	3.314323	84.041105
13	10.245005	2.390095	87.3649
14	8.098875	1.843482	90.057643
15	6.645834	1.27306	92.081106
16	5.100367	0.836735	94.062898
17	4.151604	0.849365	94.99903
18	3.494187	0.566537	95.939275
19	2.764277	0.447117	96.788606
20	2.035935	0.286126	97.677938
21	1.475254	0.235169	98.289578
22	1.298565	0.137296	98.564139
23	1.077146	0.172967	98.749887
24	0.838435	0.134519	99.027046
25	0.622845	0.093262	99.283893

Fadi Botros
T02