Jeremy DePoyster

Professor Nicholas Penney

University of Florida

COP3503 Summer 2020 Online

Lab 3 Extra Credit

Setup:

For this project, I implemented a print function that output the type, scale, N, Push time, Pop time, and Resize counts for each operation. This allowed easy copy-and-paste to a .csv file for data analysis and simple chart-rendering.

In 'main.cpp' I built a Test method using the following structure:

```
Test( ABQ_or_ABS, N, scale_factor )
```

This way, I was able to loop through arrays of the test N's and scale factors easily in the test file. The test method would then push/enqueue and pop/dequeue the appropriate number of N's at the correct scale factor. I implemented a print method that output the type, scale, N, Push time, Pop time, and Resize counts for each operation, with each data point delimited by a comma. This allowed easy copy-and-paste to a .csv file for data analysis and simple chart-rendering.

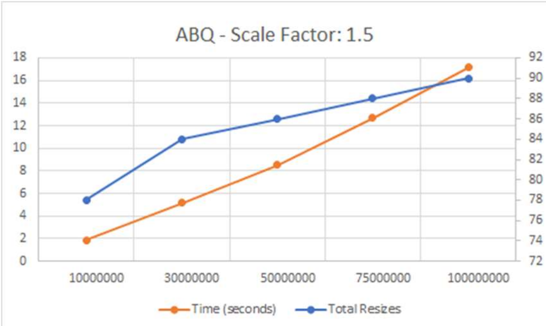
Initially, the program crashed due to the floating point comparisons of ratios in the pop / dequeue segments, but thanks to fellow student Akash Patel in the class Slack channel, I was able to create an "if" statement that would round floats within a 0.00001 difference. Similarly, my initial Lab solution was re-creating the Stack or Queue on each pop / dequeue, but I rewrote this to only re-create when a resize was in order. This was achieved using an int "counter" variable that would increment when dequeuing, using this as the "front" of the queue, rather than re-copying the entire queue each time. This allowed me to run the test with N's in the millions as required, rather than thousands, also accomplished by utilizing x64 rather than x86 debugging in Visual Studio.

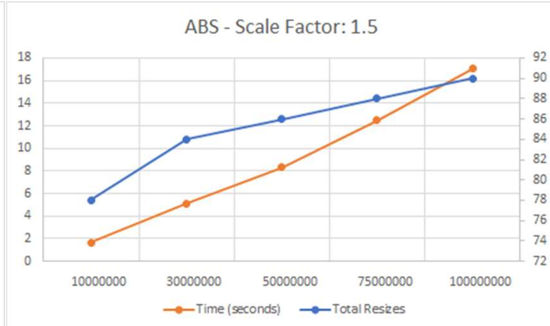
Findings:

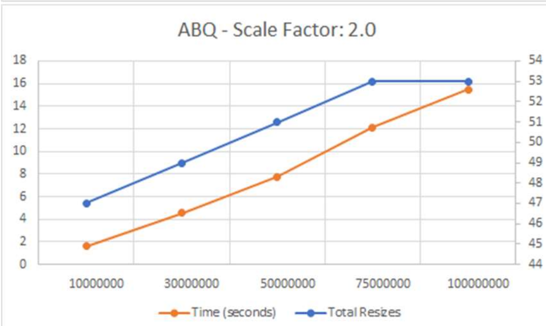
1) Based on the charts, the higher the N, the longer the time, but it seems to be on an almost direct trajectory, with a similar slope regardless of which structure was used. I think that because of the timing lines from the chart, both the ABQ and ABS are likely O(n) in Big-O complexity because of their consistent slope regardless of N input.
2) Changing the scale factor had a significant impact on how many resizes were necessary, ranging from 90 resizes at a 1.5 scale factor with 100 million N's, to just 9 resizes at the 100 scale factor with 100 million N's.
   Interestingly, changing scale factor did not have a large impact on the time necessary to push and pop large numbers of integers. At scale factor 1.5 the time required varied from nearly 2 seconds to 17 seconds in both models, while a scale factor of 100 only offered less than half of a second to a second in savings on both models. This indicates that N is a much bigger factor than scale factor in time required to render.

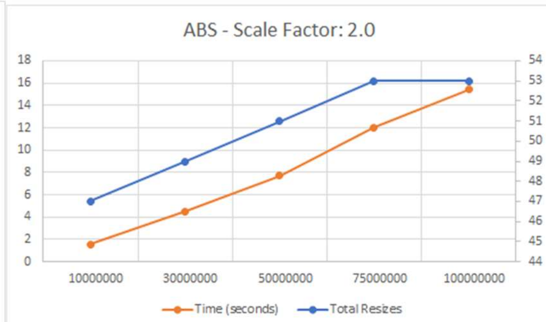
3) At the lower scale factors of 1.5 and 2, were pretty evenly trending upwards, but once reaching scale factors of 10 and 100, the number of resizes were not only largely reduced (from 50-90 down to just 9 resizes at 100), but the higher scale factors saw no difference in the number of resizes between N sizes. I think that a significantly larger N size than tested would have been necessary to see a higher scale factor change resizes.
4) The best scale factor in both models seems to be 100. This is because it requires less resizes overall, as well as nearly 2 seconds of savings on the higher end. This savings should be even more significant as more N's are introduced, as long as the memory is not necessary for other operations. In that case, 10 could be the best choice?
5) Unless I did something incorrectly, both my ABS and ABQ had very similar characteristics, even nearly identical, in this testing. Perhaps this indicates that the processing required for popping off the top or dequeuing from the front is similar, and other factors like the scale factor have a bigger impact on performance.

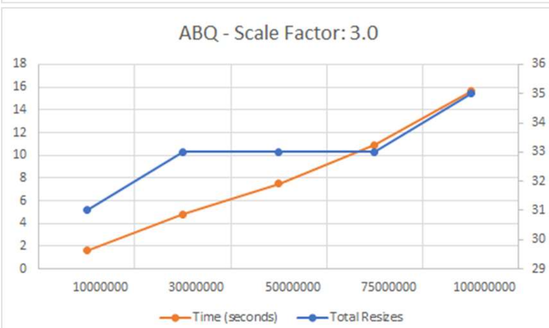
Charts for reference are accompanied on the following pages.

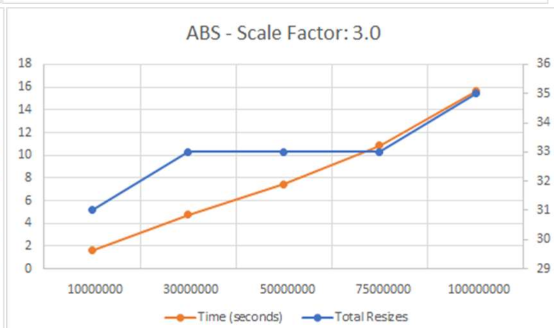

ABQ - Scale Factor: 1.5
0 2 4 6 8 10 12 14 16 18
72 74 76 78 80 82 84 86 88 90 92
10000000 30000000 50000000 75000000 100000000
Time (seconds)
Total Resizes

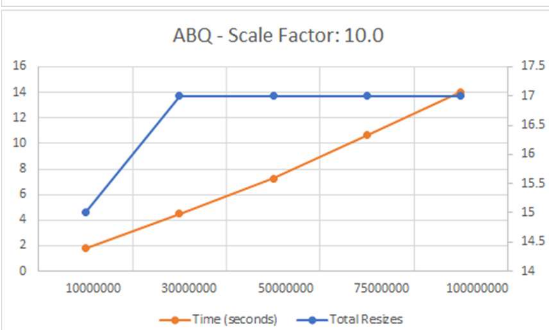

ABS - Scale Factor: 1.5
0 2 4 6 8 10 12 14 16 18
72 74 76 78 80 82 84 86 88 90 92
10000000 30000000 50000000 75000000 100000000
Time (seconds)
Total Resizes

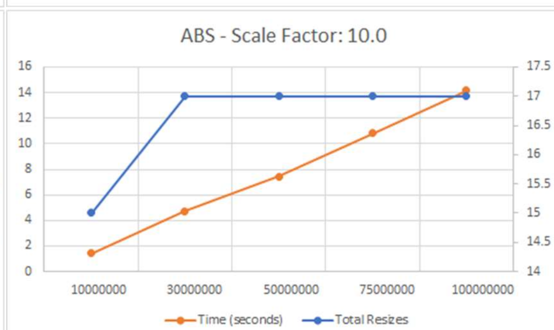

ABQ - Scale Factor: 2.0
0 2 4 6 8 10 12 14 16 18
44 45 46 47 48 49 50 51 52 53 54
10000000 30000000 50000000 75000000 100000000
Time (seconds)
Total Resizes

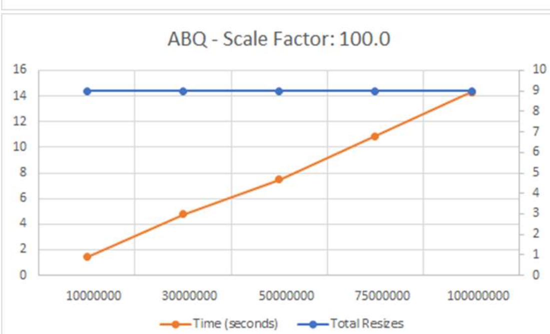

ABS - Scale Factor: 2.0
0 2 4 6 8 10 12 14 16 18
44 45 46 47 48 49 50 51 52 53 54
10000000 30000000 50000000 75000000 100000000
Time (seconds)
Total Resizes

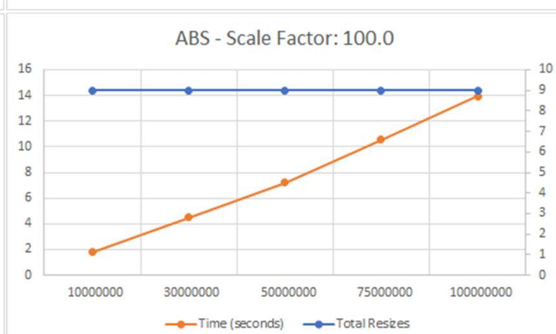

ABQ - Scale Factor: 3.0
0 2 4 6 8 10 12 14 16 18
29 30 31 32 33 34 35 36
10000000 30000000 50000000 75000000 100000000
Time (seconds)
Total Resizes


ABS - Scale Factor: 3.0
0 2 4 6 8 10 12 14 16 18
29 30 31 32 33 34 35 36
10000000 30000000 50000000 75000000 100000000
Time (seconds)
Total Resizes


ABQ - Scale Factor: 10.0
0 2 4 6 8 10 12 14 16
14 14.5 15 15.5 16 16.5 17 17.5
10000000 30000000 50000000 75000000 100000000
Time (seconds)
Total Resizes


ABS - Scale Factor: 10.0
0 2 4 6 8 10 12 14 16
14 14.5 15 15.5 16 16.5 17 17.5
10000000 30000000 50000000 75000000 100000000
Time (seconds)
Total Resizes


ABQ - Scale Factor: 100.0
0 2 4 6 8 10 12 14 16
0 1 2 3 4 5 6 7 8 9 10
10000000 30000000 50000000 75000000 100000000
Time (seconds)
Total Resizes


ABS - Scale Factor: 100.0
0 2 4 6 8 10 12 14 16
0 1 2 3 4 5 6 7 8 9 10
10000000 30000000 50000000 75000000 100000000
Time (seconds)
Total Resizes

TYPE	Scale	N	Push Time	Push Resize	Pop Time	Total Resize	Time (s)
ABQ	1.5	10000000	0.607698	39	1.25545	78	1.863148
ABQ	1.5	30000000	1.64378	42	3.52389	84	5.16767
ABQ	1.5	50000000	2.71531	43	5.79876	86	8.51407
ABQ	1.5	75000000	4.00854	44	8.65843	88	12.66697
ABQ	1.5	100000000	5.54363	45	11.5973	90	17.14093
ABQ	2	10000000	0.491324	23	1.09854	47	1.589864
ABQ	2	30000000	1.36257	24	3.18415	49	4.54672
ABQ	2	50000000	2.36746	25	5.376	51	7.74346
ABQ	2	75000000	3.80013	26	8.32854	53	12.12867
ABQ	2	100000000	4.73438	26	10.7633	53	15.49768
ABQ	3	10000000	0.50115	15	1.0901	31	1.59125
ABQ	3	30000000	1.53164	16	3.29447	33	4.82611
ABQ	3	50000000	2.24392	16	5.26026	33	7.50418
ABQ	3	75000000	3.18746	16	7.72362	33	10.91108
ABQ	3	100000000	4.84344	17	10.7784	35	15.62184
ABQ	10	10000000	0.412106	7	1.01544	15	1.427546
ABQ	10	30000000	1.52158	8	3.24813	17	4.76971
ABQ	10	50000000	2.26349	8	5.2249	17	7.48839
ABQ	10	75000000	3.2096	8	7.6753	17	10.8849
ABQ	10	100000000	4.13836	8	10.1479	17	14.28626
ABQ	100	10000000	0.64971	4	1.15115	9	1.80086
ABQ	100	30000000	1.39766	4	3.12295	9	4.52061
ABQ	100	50000000	2.14596	4	5.11226	9	7.25822
ABQ	100	75000000	3.0846	4	7.56465	9	10.64925
ABQ	100	100000000	4.01634	4	10.0298	9	14.04614
ABS	1.5	10000000	0.521741	39	1.12018	78	1.641921
ABS	1.5	30000000	1.65443	42	3.45478	84	5.10921
ABS	1.5	50000000	2.65209	43	5.67908	86	8.33117
ABS	1.5	75000000	3.99111	44	8.51123	88	12.50234
ABS	1.5	100000000	5.57134	45	11.5075	90	17.07884
ABS	2	10000000	0.484664	23	1.09549	47	1.580154
ABS	2	30000000	1.35849	24	3.15181	49	4.5103
ABS	2	50000000	2.3488	25	5.34981	51	7.69861
ABS	2	75000000	3.78491	26	8.24349	53	12.0284
ABS	2	100000000	4.73059	26	10.7201	53	15.45069
ABS	3	10000000	0.497987	15	1.08934	31	1.587327
ABS	3	30000000	1.4709	16	3.26769	33	4.73859
ABS	3	50000000	2.22614	16	5.23074	33	7.45688
ABS	3	75000000	3.17647	16	7.65614	33	10.83261
ABS	3	100000000	4.8352	17	10.7725	35	15.6077
ABS	10	10000000	0.410893	7	1.00833	15	1.419223
ABS	10	30000000	1.50542	8	3.23131	17	4.73673
ABS	10	50000000	2.24832	8	5.18173	17	7.43005
ABS	10	75000000	3.18085	8	7.63482	17	10.81567
ABS	10	100000000	4.11067	8	10.066	17	14.17667
ABS	100	10000000	0.650177	4	1.14626	9	1.796437
ABS	100	30000000	1.38813	4	3.10003	9	4.48816
ABS	100	50000000	2.14321	4	5.05629	9	7.1995
ABS	100	75000000	3.05749	4	7.50105	9	10.55854
ABS	100	100000000	3.99916	4	9.94571	9	13.94487