

Part One: Analysing and Measuring Queues

(a) Cost & BigO

Priority Queue:

Initialise queue with n patients:	times = 1	cost = $O(1)$
Repeat 100,000 times:		
Dequeue a patient from queue	times = 100,000	cost = $O(\log(n))$
Enqueue a new patient to queue	times = 100,000	cost = $O(\log(n))$
Total cost:		$O(\log(n))$

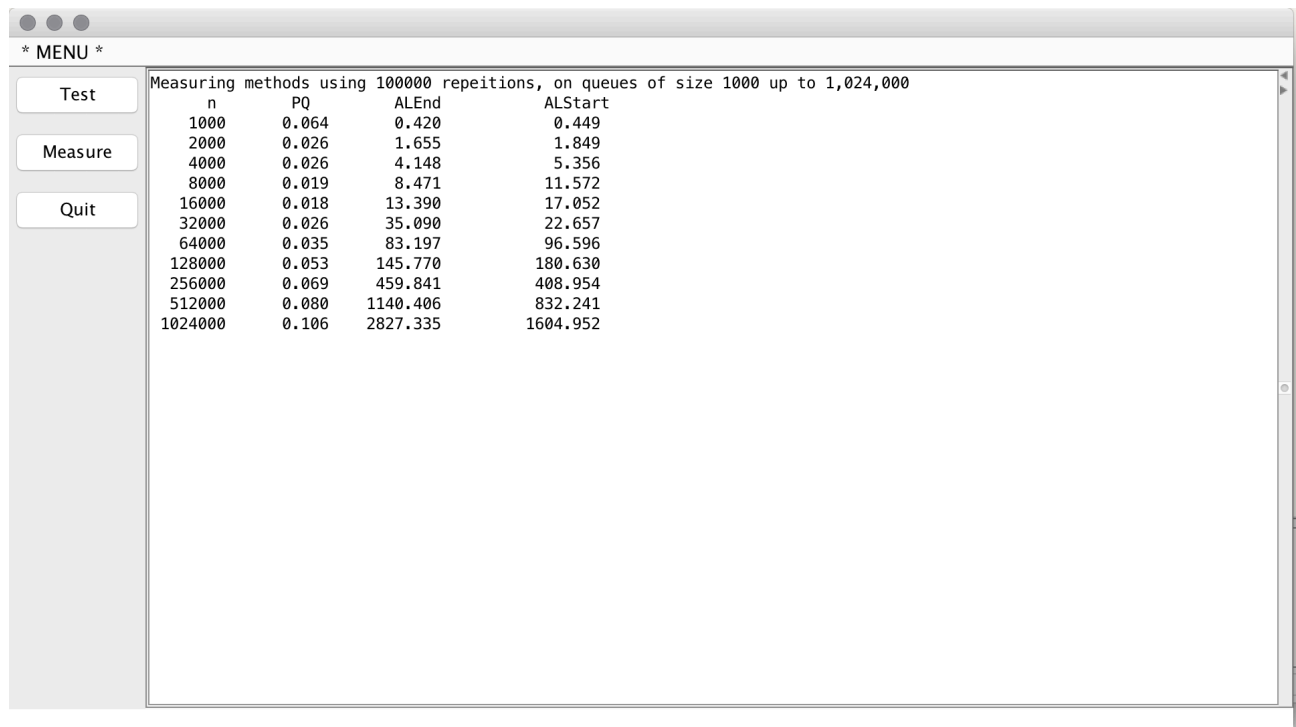
ArrayList, head at the start:

Initialise list with n patients:	times = 1	$O(n)$
Sort list	times = 1	$O(n\log(n))$
Repeat 100,000 times:		
Remove patient from list	times = 100,000	$O(n)$
Add new patient to list	times = 100,000	$O(n\log(n))$
Sort list after added	times = 100,000	$O(n)$
Total cost:		$O(n\log(n))$

ArrayList, head at the end:

Initialise list with n patients:	times = 1	$O(n)$
Sort list	times = 1	$O(n\log(n))$
Repeat 100,000 times:		
Remove patient from list	times = 100,000	$O(1)$
Add new patient to list	times = 100,000	$O(1)$
Sort list after added	times = 100,000	$O(n\log(n))$
Total cost:		$O(n\log(n))$

(b)



n	PQ	ALEnd	ALStart
1000	0.064	0.420	0.449
2000	0.026	1.655	1.849
4000	0.026	4.148	5.356
8000	0.019	8.471	11.572
16000	0.018	13.390	17.052
32000	0.026	35.090	22.657
64000	0.035	83.197	96.596
128000	0.053	145.770	180.630
256000	0.069	459.841	408.954
512000	0.080	1140.406	832.241
1024000	0.106	2827.335	1604.952

(c)

From my observations, the measuring time of PriorityQueue is quiet fast than other two ways. When the test is very short, there seems no huge differences. But when time goes increasing, PQ didn't change a lot, but ALEnd and ALStart took very very long time. It really took a long time for me to wait for the results. By the way, the measuring time of ArrayList from head is close to it of ArrayList from end. Although the results show a increasing trend but at some particular points, the measuring time had a little bit fluctuation. There is a difference between results of ALEnd and ALStart, it might be due to the statements' differences during these two method. So I think the algorithm of PQ doesn't cost too much more when running time goes up. But ArrayList cost large higher than PriorityQueue.