CSCE 313 Programming Assignment 1

Hunter Cleary - hncleary - 625001547

January 2019

1 Buddy Allocator

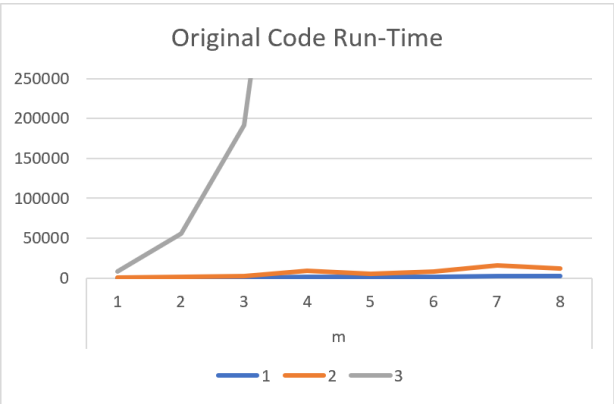
Run-time in microseconds was recorded for both the original code and the completed project. Conditions were set for each possibility within $1 < n < 3$ and $1 < m < 8$.

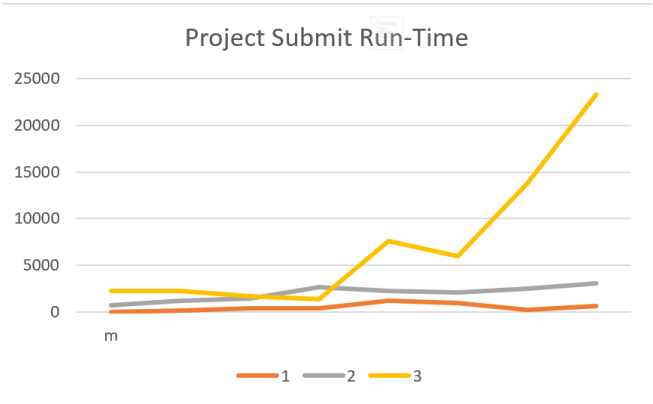
Table 1: Original Code Run Time

	m							
n	1	2	3	4	5	6	7	8
1	14	143	394	1329	1369	1236	2372	2697
2	304	1262	2507	8965	5163	7936	15865	12363
3	8524	55711	192057	767431	2987555	12168757	49822571	194735448

Table 2: Project Code Run-time

	m							
n	1	2	3	4	5	6	7	8
1	35	132	405	408	1228	999	272	616
2	722	1246	1466	2656	2257	2095	2484	3044
3	2241	2258	1730	1352	7610	5970	13735	23286





There was a clear and significant improvement in run-time when the code was changed. As demonstrated in the graphs and data above, the scale of time taken decreased drastically when the buddy allocator was more properly implemented with an efficient linked list. As allocate and free calls increased, generally, so did the time taken to run. This was especially so when n was ≥ 3. The run-time increased exponentially in both testing cases, but significantly less so in the improved version, demonstrating that the number of *alloc*() calls could more drastically change the run-time value.

The implementation of the BlockHeader linked list lead to an improved run-time. This allowed the allocator to insert, remove, merge, and split blocks without having to iterate through the entire linked list. Using block member variables, nodes could be found during recursion, allowing functions to be run effectively.