

The objective of this report is to provide an overview of my malloc program that includes my custom implementation of first fit, best fit, next fit, and worst fit algorithms, as well as functions for malloc, free, and realloc. It compares the performance of all these algorithms in terms of speed, max heap use, and relative number of splits and growth. However, during testing my algorithms did not provide the correct memory addresses for the allocated blocks.

First fit searches the list for the first block that is big enough to accommodate the requested memory. Best Fit searches the list for the smallest block that can hold the requested memory. Next Fit searches the list from the block it previously allocated for a block to hold the requested memory. Worst Fit searches the list for the biggest block that can fulfill the requested memory.

The test implementation program's objective was to evaluate my functions against the standard library's version of malloc, realloc, and free. It evaluates the implementations based on factors such as allocation time, deallocation time, memory fragmentation, and overall program performance. Through this testing process, the goal is to identify the most efficient and effective implementation for various types of programs and workloads.

Figure 1				
stats	BF	WF	NF	FF
mallocs	7	7	12	12
frees	2	2	3	3
reuses	0	0	0	0
grows	6	6	12	10
splits	0	0	0	0
coalesces	0	0	0	0
blocks	0	5	11	9

requested	13	13	24	22
max heap	12	12	24	20
time(s)	0.006s	0.005s	0.006s	0.009s
Figure 2				
	BF	WF	NF	FF
time(s)	0.007	0.003	0.003	0.003
Figure 3				
	Test5 (A)	Test5 (B)		
time(s)	0.008	0.005		

Figure 1 shows that the stats and time difference between these algorithms is negligible, but Best Fit and Next Fit request the least memory and WF and FF request and use the most memory. Figure 2 presents the results of test cases that measure the execution time of the different algorithms. The time taken by all the algorithms is similar. Figure 3 shows the results of a test case that was run using the custom malloc and free functions implemented in the program (Test5 A) and the standard library malloc (Test5 B). The execution times for both versions are similar.