REPORT

Executive Summary:

The report compares four memory allocation methods—First Fit, Next Fit, Best Fit, and Worst Fit—with the system malloc function. It examines their time performance and various memory management statistics, such as mallocs, frees, reuses, grows, splits, coalesces, blocks, requested memory, and maximum heap size, highlighting differences between them. These insights offer valuable guidance in optimizing memory management implementations.

Description of the algorithms implemented:

The implemented memory allocation algorithms include:

1. First Fit:

This method scans the memory heap to find the first available block that can accommodate the requested memory size. It swiftly locates a suitable block for allocation, but it may cause fragmentation as it chooses the first block that meets the size criteria.

2. Next Fit:

Next Fit operates similarly to First Fit but starts the search for a free block from the position of the last allocated block. Its aim is to reduce fragmentation by potentially filling in the gaps left by previously allocated blocks.

3. Best Fit:

Best Fit examines the entire memory heap to locate the smallest available block that can fit the requested memory size. It optimizes memory usage by selecting the block that minimizes fragmentation, albeit at the expense of longer search times.

4. Worst Fit:

In contrast to Best Fit, Worst Fit chooses the largest available block that can hold the requested memory. This method aims to minimize the creation of small free blocks, potentially reducing fragmentation over time.

Each algorithm has its own balance of advantages and drawbacks regarding time complexity, memory fragmentation, and overall performance. These characteristics are evaluated through practical testing and analysis in the report.

Test Implementation:

To decide on the best way to test the memory allocation algorithms, I went through few trials:

1. Trying Individual Tests:

I thought about doing separate tests for each algorithm. But when I did this, I found that the results were not consistent. It was hard to make clear conclusions.

2. Thinking About One Big Test:

I also thought about doing just one test for all four algorithms together. But this didn't work well because it was difficult to see the differences between the algorithms.

3. Choosing Two Tests:

Two tests were chosen: First Fit vs. Next Fit, and Best Fit vs. Worst Fit. In each test, I allocated different-sized memory blocks, recorded details like addresses and times, and analysed how well each algorithm managed memory.

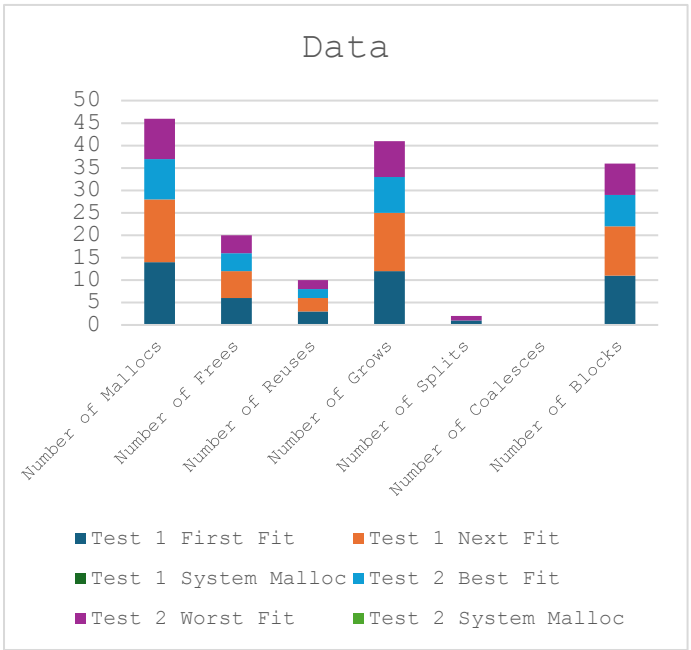
Advantages:

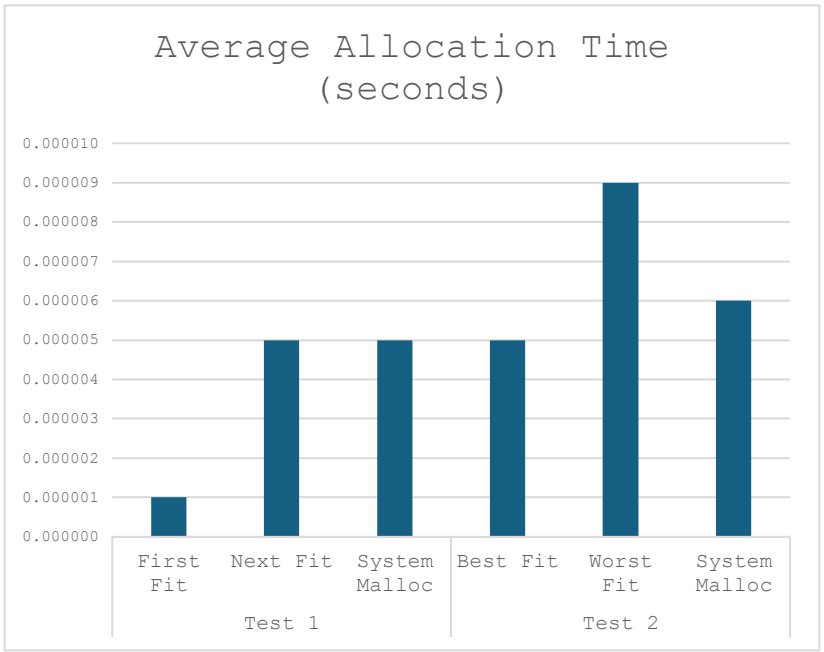
- Splitting the testing into two parts made it more organized and easier to compare the algorithms.
- This approach gives clearer insights into how each algorithm works and helped in making better conclusions.

Test Results:

In analyzing the test results, distinct differences emerged between the memory allocation algorithms and the performance of the system malloc function. Here's a breakdown of the findings:

Test Case	Algorithm	Avg. Time	No. Mallocs	No. Frees	No. Reuses	No. Grows	No. Splits	No. Coalesces	No. Blocks
Test 1	First Fit	0.000001	14	6	3	12	1	0	11
	Next Fit	0.000005	14	6	3	13	0	0	11
	System Malloc	0.000005	-	-	-	-	-	-	-
Test 2	Best Fit	0.000005	9	4	2	8	0	0	7
	Worst Fit	0.000009	9	4	2	8	1	0	7
	Sys. Malloc	0.000006	-	-	-	-	-	-	-





Interpretations:

Both First Fit and Next Fit algorithms showed similar performance in terms of allocation time and heap management statistics. Heap management statistics indicate moderate fragmentation and consistent behavior between the two algorithms.

Both Best Fit and Worst Fit algorithms showed comparable performance in terms of allocation time and heap management statistics. Despite the split observed in Worst Fit, its overall behavior closely resembled that of Best Fit, indicating similar memory utilization patterns.

Comparison with System Malloc:

Heap management statistics showed minor variations compared to the algorithms, suggesting differences in internal memory management strategies between the system malloc and the tested algorithms.

Anomaly Detected:

The most notable anomaly was observed in the performance of the Worst Fit algorithm, which took a significantly longer time (0.000009 seconds) compared to other algorithms. This discrepancy suggests potential inefficiencies in Worst Fit.

Conclusion:

In conclusion, the comparison of memory allocation algorithms highlights their varied performance and implications for system efficiency. While some algorithms exhibit similar behavior, subtle differences underscore the importance of optimizing memory management strategies.