**Homework 4 – Heap Assignment**

**Name:** Goutami Padmanabhan **UTA ID:** 1001669338

**Benchmarks: A discussion on how you benchmarked your library and what the results were.**

Which heap management strategy does the best job of reusing free blocks? Best Fit

Which one is the worst? Next Fit

Which heap management strategy requires the least amount of heap space? Best Fit

Which one is the worst? Worst Fit

Which heap management strategy allows for the

most splits? Worst Fit

most coalescing? Worst Fit

least splits? Best Fit

least coalescing? Best Fit

Which heap management strategy was the

fastest? Next Fit

slowest? Best Fit

Consider the benchmark programs.

Which one requires the most

mallocs? Next Fit

frees? Next Fit

Which one requests the most amount of space? Worst Fit

Which one requires the largest heap? Worst Fit

Which heap management strategy suffers the most from fragmentation (what type)? First Fit – Internal fragmentation

Which heap management strategy is the best? Next Fit

**Summary:**

**First Fit**:

Concept: First fit allocates a process to a block which is the first enough block that is able to accommodate the process. It starts searching from the first block.

***Advantages***: Simple and fast search

***Disadvantages***: Internal fragmentation at the beginning of the search. Causes External Fragmentation also. Wastage of space.

**Next Fit**:

Concept: Next Fit allocates a process to a block which is the next enough block that is able to accommodate the process. It starts searching from the last block it left previously.

***Advantages***: Avoids internal fragmentation by searching from where it ends last time. Fastest search.

**Best Fit**:

Concept: Best fit keeps searching for the minimum block size available that could accommodate a process and allocates it.

***Advantages***: Space is efficiently utilized. Memory utilization is the best.

***Disadvantages***: It consumes a lot of time to search for the best block to accommodate. Slow search. As a result of this, performance is less.

**Worst Fit**: Worst fit keeps searching for the maximum block size available that could accommodate a process and allocates it.

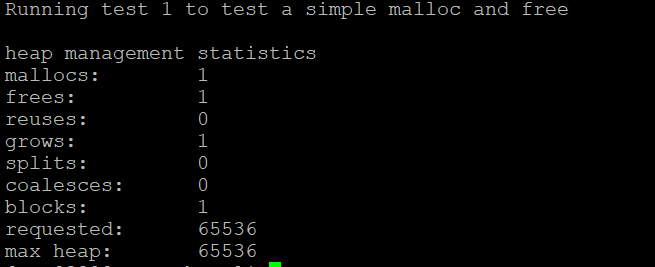
***Advantages***: It reduces the small holes in memory.

***Disadvantages***: Larger block sizes are occupied at the beginning. If a process requires large space at the end, it is incapable of providing the space.

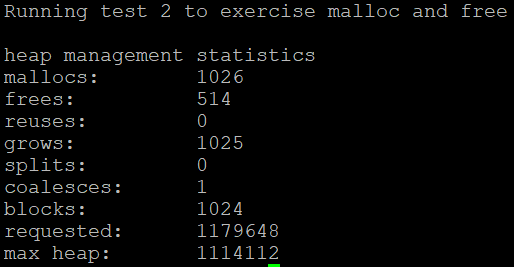
**Results for Analysis:**

**First Fit**

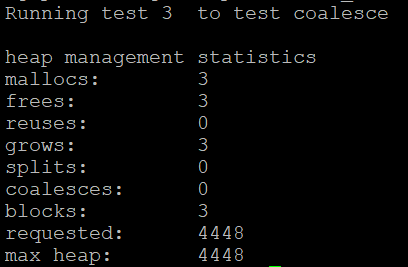
Test1.c



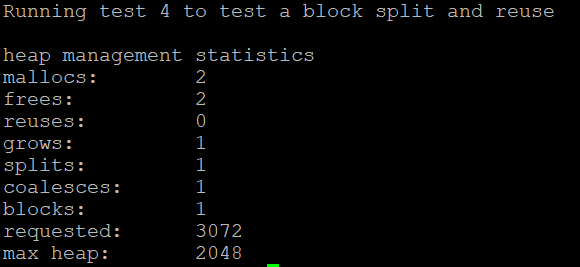
Test2.c



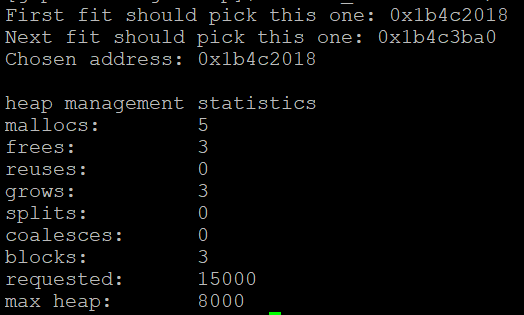
Test3.c



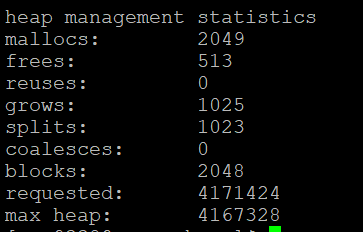
Test4.c



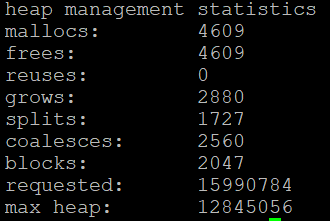
Ffnf.c



Bench1.c

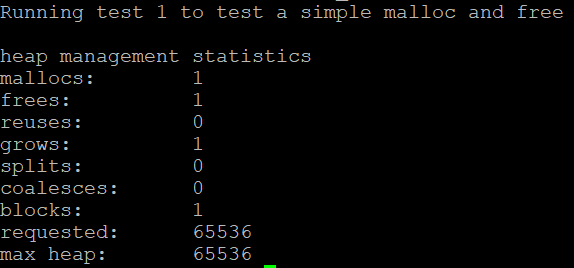


Bench2.c

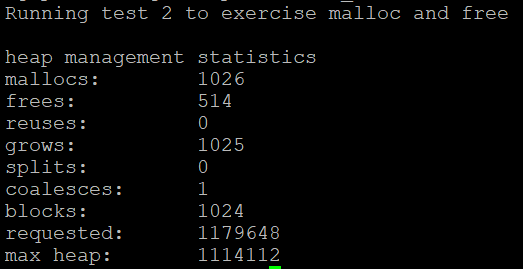


**Next Fit:**

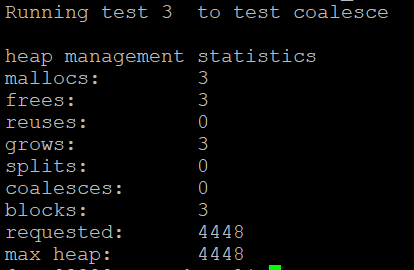
Test1.c



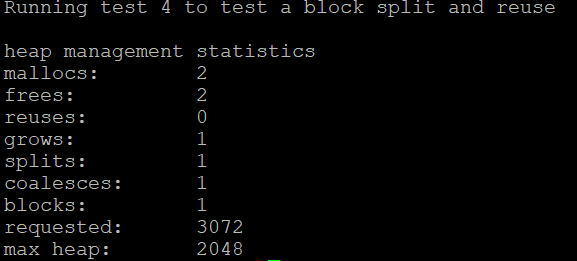
Test2.c



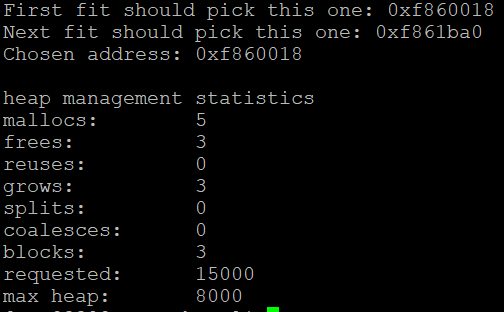
Test3.c



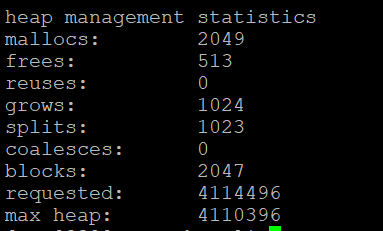
Test4.c



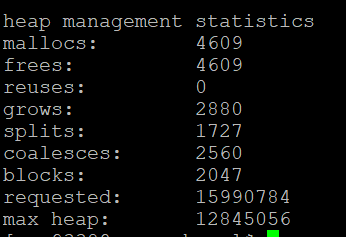
Ffnf.c



Bench1.c

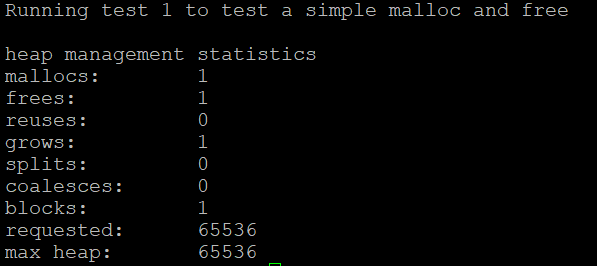


Bench2.c

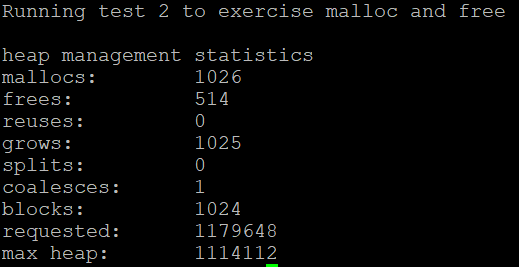


**Best Fit:**

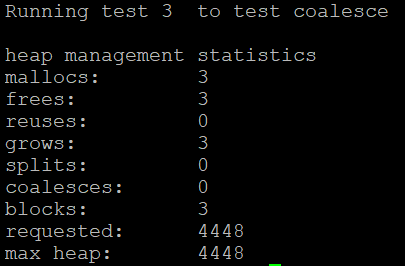
Test1.c



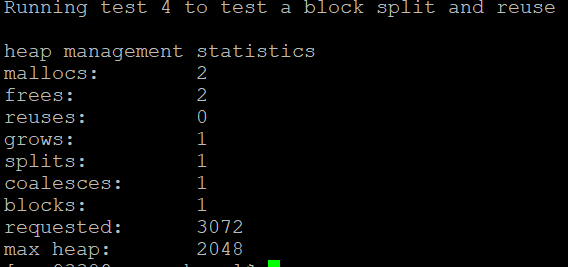
Test2.c



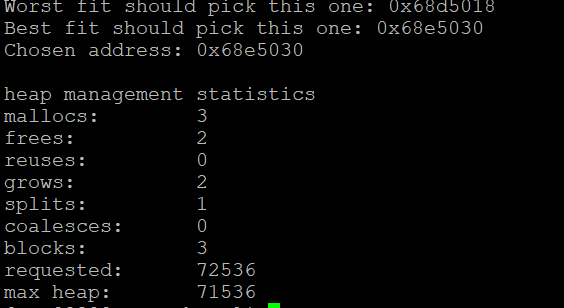
Test3.c



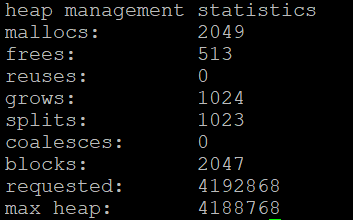
Test4.c



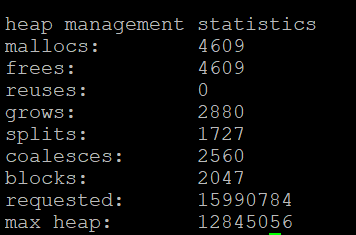
Bfwf.c



Bench1.c

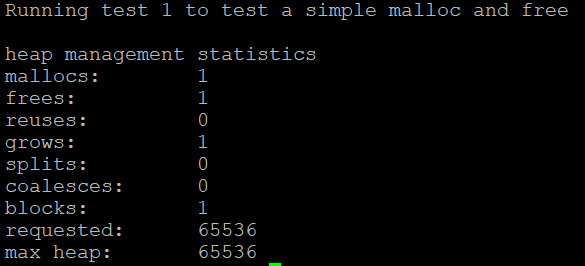


Bench2.c

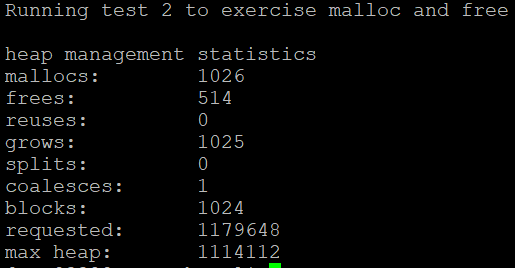


**Worst Fit:**

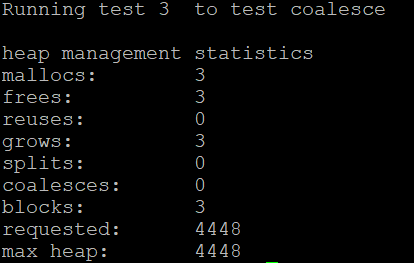
Test1.c



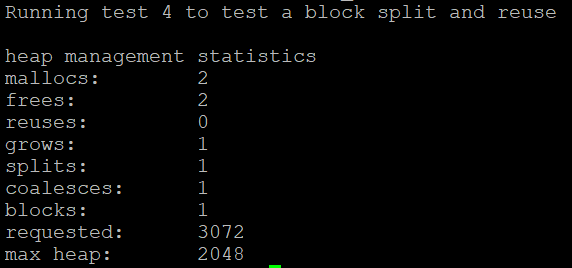
Test2.c



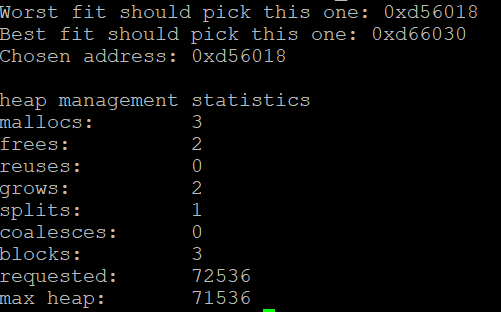
Test3.c



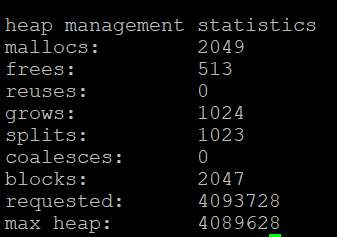
Test4.c



Bfwf.c



Bench1.c



Bench2.c

