PROGRAMMING ASSIGNMENT 4

HEAP MANAGEMENT

GROUP MEMBERS:

Pratik Barhate: 1001649826
Darshak Rairakhia: 1001637657

BENCHMARKS-

We first wrote the code on omega and then checked all the test files to check if all the heap management algorithms were properly working or not. After doing that we benchmarked our library for all the heap management algorithms which are first fit, next fit, best fit and the worst fit.

And these are the management strategies we got after benchmarking our libraries:

```
[prb9826@omega Heap-master]$ env LD_PRELOAD=lib/libmalloc-ff.so tests/bench1

heap management statistics
mallocs:        2049
frees:          513
reuses:         2
grows:          2047
splits:         0
coalesces:      0
blocks:         2047
requested:      4207900
max heap:       4206088
```

```
[prb9826@omega Heap-master]$ env LD_PRELOAD=lib/libmalloc-nf.so tests/bench1

heap management statistics
mallocs:        2049
frees:          513
reuses:         2
grows:          2047
splits:         0
coalesces:      0
blocks:         2047
requested:      4257300
max heap:       4254800
```

```
[prb9826@omega Heap-master]$ env LD_PRELOAD=lib/libmalloc-bf.so tests/bench1

heap management statistics
mallocs:        2049
frees:          513
reuses:         2
grows:          2047
splits:         0
coalesces:      0
blocks:         2047
requested:      4298276
max heap:       4295600
```

```
[prb9826@omega Heap-master]$ env LD_PRELOAD=lib/libmalloc-wf.so tests/bench1

heap management statistics
mallocs:        2049
frees:          513
reuses:         3
grows:          2046
splits:         0
coalesces:      0
blocks:         2046
requested:      4267928
max heap:       4263976
```

```
[prb9826@omega Heap-master]$ env LD_PRELOAD=lib/libmalloc-ff.so tests/bench2

heap management statistics
mallocs:        4609
frees:          4609
reuses:         1538
grows:          3071
splits:         0
coalesces:      0
blocks:         3071
requested:      15990784
max heap:       14667776
[prb9826@omega Heap-master]$ env LD_PRELOAD=lib/libmalloc-bf.so tests/bench2

heap management statistics
mallocs:        4609
frees:          4609
reuses:         1538
grows:          3071
splits:         0
coalesces:      0
blocks:         3071
requested:      15990784
max heap:       14667776
[prb9826@omega Heap-master]$ env LD_PRELOAD=lib/libmalloc-wf.so tests/bench2

heap management statistics
mallocs:        4609
frees:          4609
reuses:         1538
grows:          3071
splits:         0
coalesces:      0
blocks:         3071
requested:      15990784
max heap:       14667776
[prb9826@omega Heap-master]$ env LD_PRELOAD=lib/libmalloc-nf.so tests/bench2

heap management statistics
mallocs:        4609
frees:          4609
reuses:         1538
grows:          3071
splits:         0
coalesces:      0
blocks:         3071
requested:      15990784
max heap:       14667776
[prb9826@omega Heap-master]$
```

Reusing free blocks:
The heap management strategy that does the best job of reusing free blocks is the best fit algorithm.
The heap management strategy that does the worst job of reusing free blocks is the next fit algorithm.

Amount of heap space required:
The heap management strategy that requires the least amount of heap space is the best fit algorithm.
The heap management strategy that requires the most amount of heap space is the worst fit algorithm.

Amount of Coalescing:
The heap management strategy that requires the most amount of coalescing is the best fit algorithm.
The heap management strategy that requires the least amount of coalescing is the worst fit algorithm.

Fastest Strategy:
The heap management strategy that is the fastest is the next fit algorithm.
The heap management strategy that is the slowest is the best fit algorithm.

Considering the benchmark problems:

- Next fit requires the maximum number of mallocs.
- Worst fit requires the most amount of space.
- Next fit requires the largest heap.

ANALYSIS:

The heap management strategy that suffers the most from the fragmentation is the first fit algorithm. It suffers from external fragmentation

The best heap management strategy is the best fit algorithm because it finds the fits that are only a few bytes larger. So now because of this the fragmentation is also only a few bytes. Leaving large blocks untouched will also help the cache locality and minimizes the load. A good best fit algorithm will maintain its speed even while managing large number of small fragments.

SUMMARY:

So what we learned about memory management is that the best strategy to give a process a particular free block should be very efficient and by that we mean that it should be considerably fast to execute and it should suffer from low fragmentation. The best strategy should work with all these requirements.