OS Project – 3

Turgut Alp Edis – 21702587

Doğukan Ertunga Kurnaz – 21702331

Experiments:

We run experiments for the following categories:

1) Random Allocation sizes when the segment size is fixed to 2^{20}
2) Fixed allocation sizes when the segment size changes

1) Random Allocation Sizes When the Segment Size is Fixed to 2^{20}

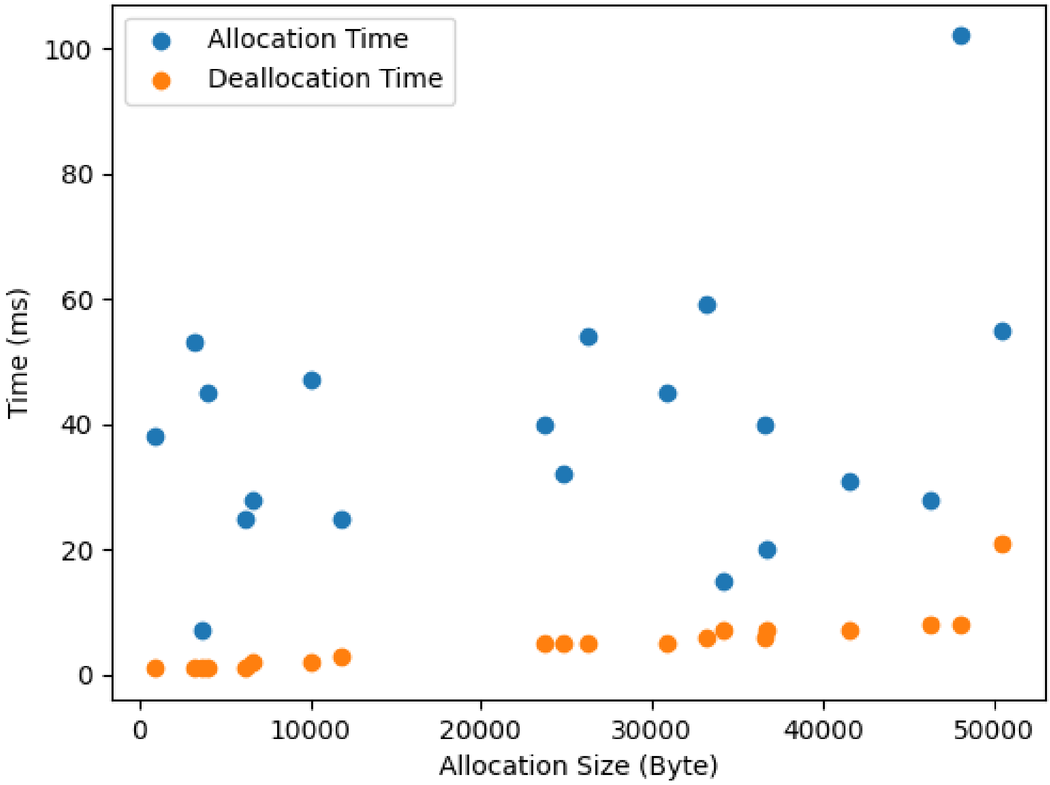

Figure 1. Time of allocation and deallocation when the sizes are random

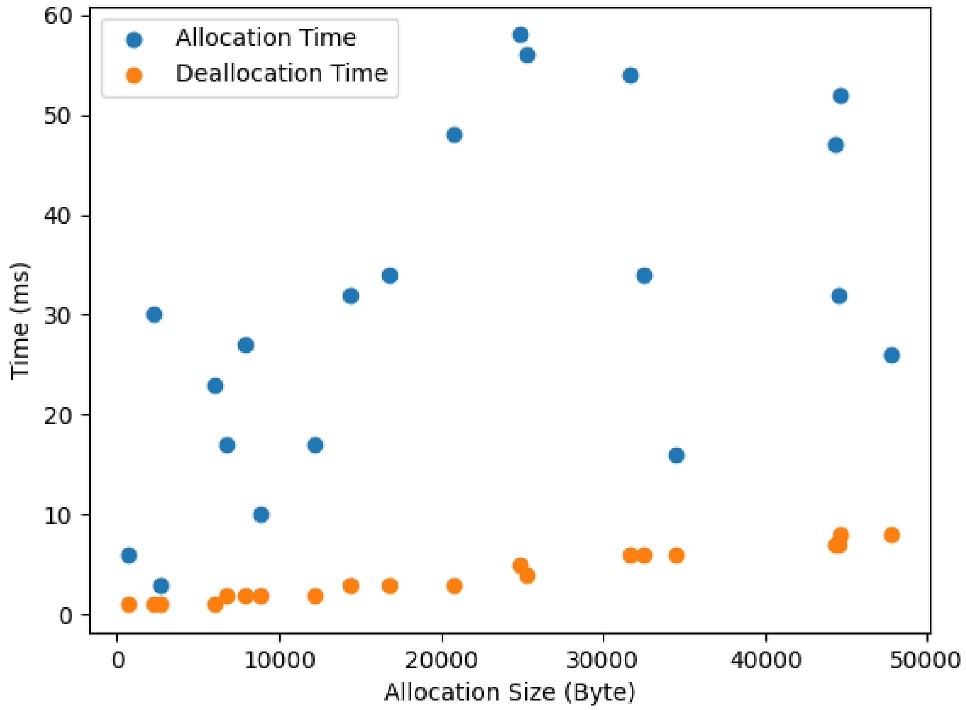

Figure 2. Time of allocation and deallocation when the sizes are random

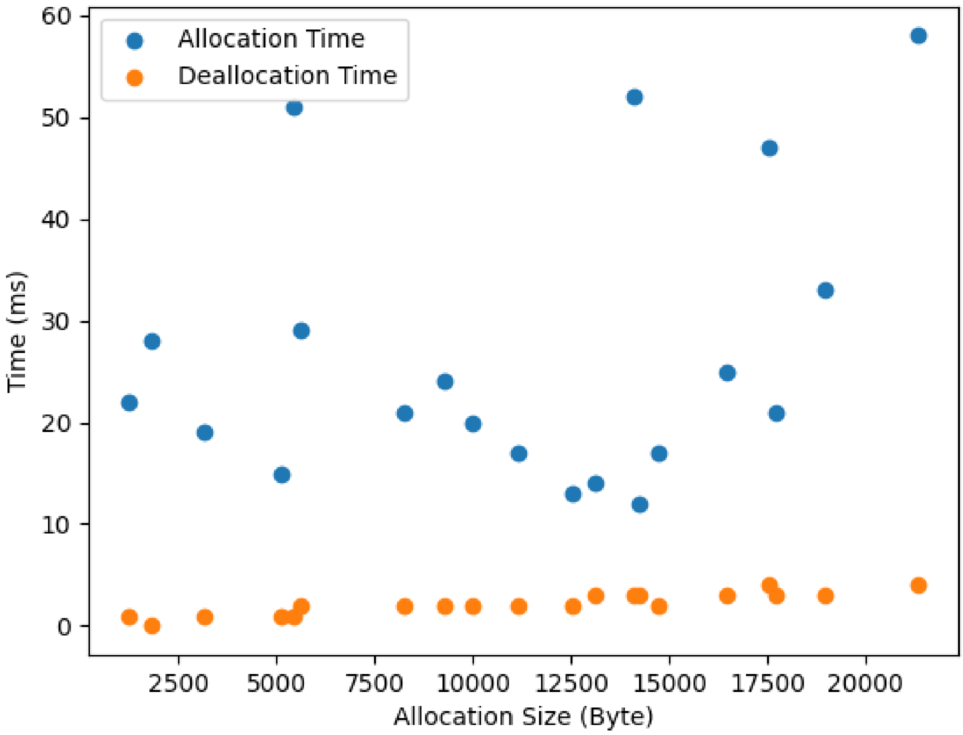

Figure 3. Time of allocation and deallocation when the sizes are random

From these 3 figures, it can be said that the sequence of the bytes is important because they are randomly arranged and regardless of the size, some allocations took longer than others. The reason is that their allocation comes after all other allocations.

Deallocations are close to each other since the information of byte is already known and only job is to delete the bitmap location and deallocate from the memory segment. However, for allocations, it should search the bitmap to find the place that fits first. Therefore, it takes much longer than deallocation.

For internal fragmentation, the amounts are 109, 174 and 166, respectively. When the fragmentation is lower, the mean of the allocation time, which can be indicated from the graphs, decreases because there are less effort to find the nearest possible size for the given size.

2) Fixed Allocation Sizes When the Segment Size Changes

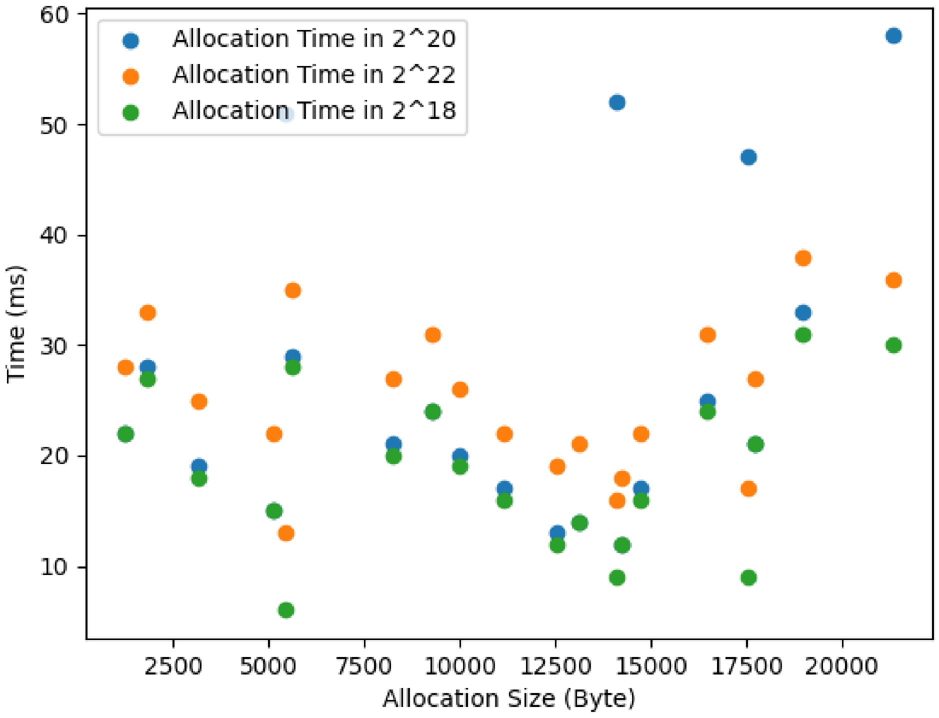

Figure 4. The allocation time comparison between different segment sizes

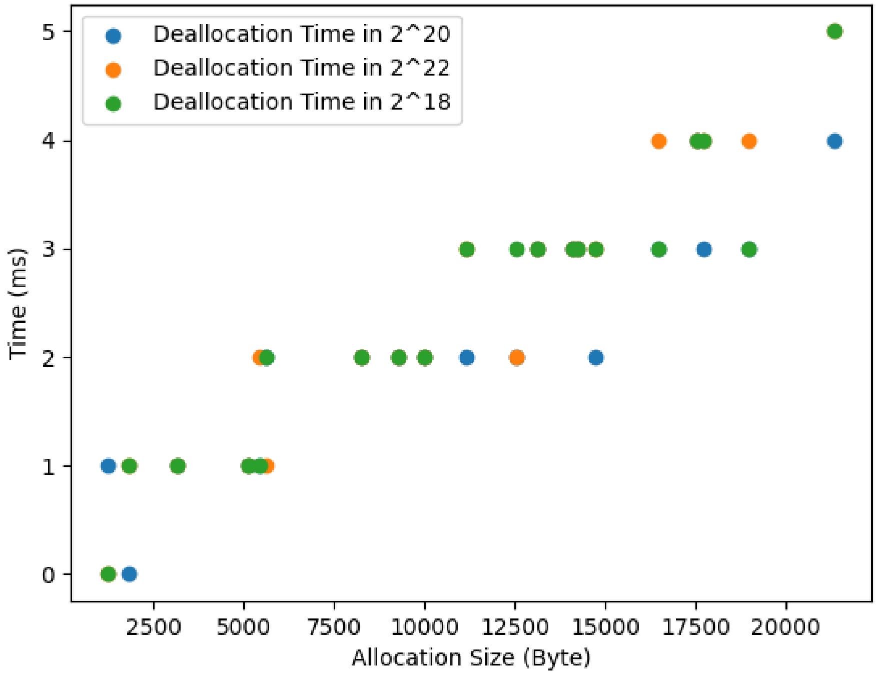

Figure 5. The deallocation time comparison between different segment sizes

From these 2 figures, it can be stated that the deallocation time does not differ much for different segment sizes. The reason is the amount of job does not dependent on the size of the segment. However, for allocation, almost all of the bytes are allocated more faster than others in the least segment size. The reason is when the segment size is lower, there are less elements in the bitmap, so the search is faster due to less elements. Therefore, when the segment size decreases, the time for allocation decreases as well.