- 姓名：杨淳瑜
- 学号：22307140114
- 日期：2024.10.21

1

`./malloc.py -n 10 -H 0 -p BEST -s 0 -c`

This command generates 10 operations, specifies a heap of 100 byte and uses BEST search policy.

Overtime, there are more segments in the free space.

```
ptr[0] = Alloc(3) returned 1000 (searched 1 elements)
Free List [ Size 1 ]: [ addr:1003 sz:97 ]

Free(ptr[0])
returned 0
Free List [ Size 2 ]: [ addr:1000 sz:3 ][ addr:1003 sz:97 ]

ptr[1] = Alloc(5) returned 1003 (searched 2 elements)
Free List [ Size 2 ]: [ addr:1000 sz:3 ][ addr:1008 sz:92 ]

Free(ptr[1])
returned 0
Free List [ Size 3 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1008 sz:92 ]

ptr[2] = Alloc(8) returned 1008 (searched 3 elements)
Free List [ Size 3 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1016 sz:84 ]

Free(ptr[2])
returned 0
Free List [ Size 4 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1008 sz:8 ][
addr:1016 sz:84 ]

ptr[3] = Alloc(8) returned 1008 (searched 4 elements)
Free List [ Size 3 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1016 sz:84 ]

Free(ptr[3])
returned 0
Free List [ Size 4 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1008 sz:8 ][
addr:1016 sz:84 ]

ptr[4] = Alloc(2) returned 1000 (searched 4 elements)
Free List [ Size 4 ]: [ addr:1002 sz:1 ][ addr:1003 sz:5 ][ addr:1008 sz:8 ][
addr:1016 sz:84 ]

ptr[5] = Alloc(7) returned 1008 (searched 4 elements)
Free List [ Size 4 ]: [ addr:1002 sz:1 ][ addr:1003 sz:5 ][ addr:1015 sz:1 ][
addr:1016 sz:84 ]
```

2

When we use WORST policy, there are even more segments in the free space. The allocated spaces are stacked in the end of address space.

```
ptr[0] = Alloc(3) returned 1000 (searched 1 elements)
Free List [ Size 1 ]: [ addr:1003 sz:97 ]

Free(ptr[0])
returned 0
Free List [ Size 2 ]: [ addr:1000 sz:3 ][ addr:1003 sz:97 ]

ptr[1] = Alloc(5) returned 1003 (searched 2 elements)
Free List [ Size 2 ]: [ addr:1000 sz:3 ][ addr:1008 sz:92 ]

Free(ptr[1])
returned 0
Free List [ Size 3 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1008 sz:92 ]

ptr[2] = Alloc(8) returned 1008 (searched 3 elements)
Free List [ Size 3 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1016 sz:84 ]

Free(ptr[2])
returned 0
Free List [ Size 4 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1008 sz:8 ][
addr:1016 sz:84 ]

ptr[3] = Alloc(8) returned 1016 (searched 4 elements)
Free List [ Size 4 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1008 sz:8 ][
addr:1024 sz:76 ]

Free(ptr[3])
returned 0
Free List [ Size 5 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1008 sz:8 ][
addr:1016 sz:8 ][ addr:1024 sz:76 ]

ptr[4] = Alloc(2) returned 1024 (searched 5 elements)
Free List [ Size 5 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1008 sz:8 ][
addr:1016 sz:8 ][ addr:1026 sz:74 ]

ptr[5] = Alloc(7) returned 1026 (searched 5 elements)
Free List [ Size 5 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1008 sz:8 ][
addr:1016 sz:8 ][ addr:1033 sz:67 ]
```

3

The number of searches for available space is reduced.

```
ptr[0] = Alloc(3) returned 1000 (searched 1 elements)
Free List [ Size 1 ]: [ addr:1003 sz:97 ]

Free(ptr[0])
returned 0
```

```
Free List [ Size 2 ]: [ addr:1000 sz:3 ][ addr:1003 sz:97 ]

ptr[1] = Alloc(5) returned 1003 (searched 2 elements)
Free List [ Size 2 ]: [ addr:1000 sz:3 ][ addr:1008 sz:92 ]

Free(ptr[1])
returned 0
Free List [ Size 3 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1008 sz:92 ]

ptr[2] = Alloc(8) returned 1008 (searched 3 elements)
Free List [ Size 3 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1016 sz:84 ]

Free(ptr[2])
returned 0
Free List [ Size 4 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1008 sz:8 ][
addr:1016 sz:84 ]

ptr[3] = Alloc(8) returned 1008 (searched 3 elements)
Free List [ Size 3 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1016 sz:84 ]

Free(ptr[3])
returned 0
Free List [ Size 4 ]: [ addr:1000 sz:3 ][ addr:1003 sz:5 ][ addr:1008 sz:8 ][
addr:1016 sz:84 ]

ptr[4] = Alloc(2) returned 1000 (searched 1 elements)
Free List [ Size 4 ]: [ addr:1002 sz:1 ][ addr:1003 sz:5 ][ addr:1008 sz:8 ][
addr:1016 sz:84 ]

ptr[5] = Alloc(7) returned 1008 (searched 3 elements)
Free List [ Size 4 ]: [ addr:1002 sz:1 ][ addr:1003 sz:5 ][ addr:1015 sz:1 ][
addr:1016 sz:84 ]
```

4 (By deduction)

Under all circumstances, the FIRST policy is the fastest, as all other policies need to search the entire free list to find a solution. SIZESORT+ works well for the BEST policy, while SIZESORT- is better suited for the WORST policy.

5

We use BEST policy for all tests and measure the segments of free space after all operations. We can see that coalescing reduces the segmentation of space, especially for ADDRSORT ordering.

	With Coalescing	Without Coalescing
ADDRSORT	1	31
SIZESORT+	28	31
SIZESORT-	33	31

6

`./malloc.py -n 1000 -H 0 -p BEST -s 3 -c` with probabilites ranging from 40 to 100

P	Segments
10	33
20	43
30	34
40	44
50	40
60	37
70	39
80	12
90	2
100	0

As the percentage of allocation operations increases, the number of segments first rises and then falls. Initially, with a smaller percentage of allocations, more allocations lead to greater segmentation across the memory space. However, as the allocation percentage grows larger, it becomes harder to find available space, causing the number of segments to decrease again.

7

```
> ./malloc.py -S 4 -H 0 -p FIRST -c -A +1,+1,+1,+1,-0,-1,-2,-3

ptr[0] = Alloc(1) returned 1000 (searched 1 elements)
Free List [ Size 1 ]: [ addr:1001 sz:3 ]

ptr[1] = Alloc(1) returned 1001 (searched 1 elements)
Free List [ Size 1 ]: [ addr:1002 sz:2 ]

ptr[2] = Alloc(1) returned 1002 (searched 1 elements)
Free List [ Size 1 ]: [ addr:1003 sz:1 ]

ptr[3] = Alloc(1) returned 1003 (searched 1 elements)
Free List [ Size 0 ]:

Free(ptr[0])
returned 0
Free List [ Size 1 ]: [ addr:1000 sz:1 ]

Free(ptr[1])
returned 0
Free List [ Size 2 ]: [ addr:1000 sz:1 ][ addr:1001 sz:1 ]
```

```
Free(ptr[2])
returned 0
Free List [ Size 3 ]: [ addr:1000 sz:1 ][ addr:1001 sz:1 ][ addr:1002 sz:1 ]

Free(ptr[3])
returned 0
Free List [ Size 4 ]: [ addr:1000 sz:1 ][ addr:1001 sz:1 ][ addr:1002 sz:1 ][
addr:1003 sz:1 ]
```

Note that coalescing solves this problem.

If we try different policies here. Bam, all the same.

Policy	**Segments**
FIRST	4
BEST	4
WORST	4