Homework 1

Problem 1

In the simple allocator example shown at the section 9.9.12, why there is one **prologue block**？ Give out your reason. (Hint: See the code of function *prev_alloc*)

Answer:

1. The prologue block is to eliminate the edge condition during coalescing.

   Because in the function **clalesce**, there is a line of code:

   size_t prev_alloc = GE_ALLOC(FTRP(PREV_BLKP(bp)));

   and

   *#define PREV_BLKP(bp) ((void *)(bp) - GET_SIZE(((void *)(bp) - DSIZE)))*

   If there is no prologue block, the PREV_BLKP of the first block has to be considered as a special case for we have no idea what GET_SIZE(((void *)(FirstBlockAddress) - DSIZE)) is.

Problem 2

Implement a find best fit function for the simple allocator in the book. it should perform a best fit search of the implicit free list. Write down your code. (if needed, add some comments)

static void *find_best_fit (size_t asize)

Example code:

We need one temp variable to store the best fit block

```
static void * find_best_fit (size_t asize)
{
    /* Best fit search */
    void *bp;
```

```
    void *best =NULL;
    usigned int min = -1 ;
    for (bp = heap_listp; GET_SIZE(HDRP(bp)) > 0; bp =
NEXT_BLKP(bp)) {
        if (!GET_ALLOC(HDRP(bp)) && (asize <= GET_SIZE(HDRP(bp)))){
            if(GET_SIZE(HDRP(bp)) == asize)
                return bp;
            if(GET_SIZE(HDRP(bp))-asize < min ){
                min = GET_SIZE(HDRP(bp)) - asize;
                best = bp;
            }
        }
    }
    return best;
}
```

Problem 3

What is a false fragmentation? And how to solve this phenomenon? Please list at least two ways and make a simple introduction about them.

Answer:

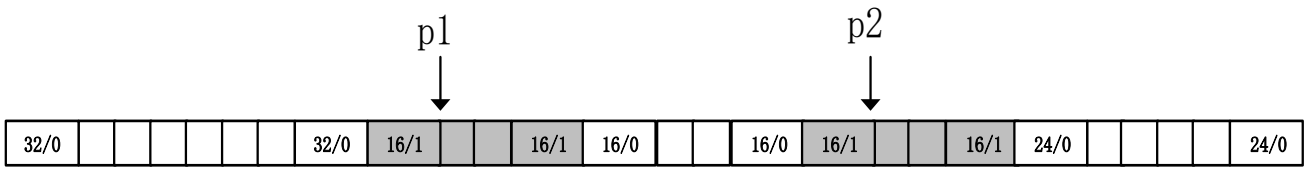
False fragmentation is a phenomenon where there is a lot of available free memory chopped up into small, unusable free blocks.

solution	advantage	disadvantage
Immediate coalescing	Coalescing fast	Not fit for repeated and frequent operations
Deferred coalescing	Fit for repeated and frequent operations	Coalescing on demand

Problem 4

The figure simulates the initial status of memory at a certain time. Allocated

blocks are shaded, and free blocks are blank (each block represents 4 bytes). The allocator maintains double-word alignment. Given the execution sequence of memory allocation operations (malloc() or free()) from 1 to 4.

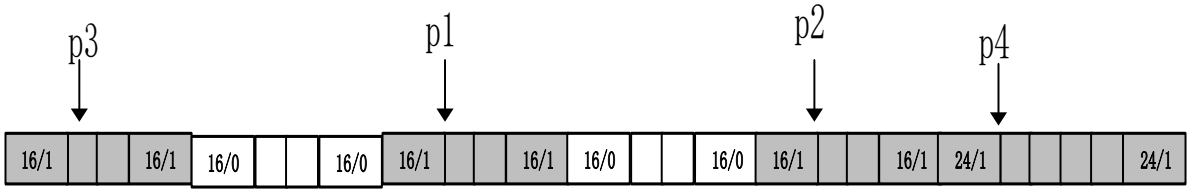


1: *P3 = malloc(4);*
2: *P4 = malloc(10);*
3: *free(P1);*
4: *P5 = malloc(16);*

Assume **first-fit** algorithm is used to find free blocks and coalesce immediately. Please draw the status of memory and mark with variables after the **2nd** and **4th** operation is executed.

Answer:

2nd



4th