15-213: Introduction to Computer Systems
Written Assignment 7

This written homework covers dynamic memory allocation.

Directions

Complete the question(s) on the following pages with single paragraph answers. These questions are not meant to be particularly long! Once you are done, submit this assignment on Canvas.

Below is an example question and answer.

Q: Please describe benefits of two's-complement signed integers versus other approaches.

```
A: Other representations of signed integers (ones-complement and
sign-and-magnitude) have two representations of zero (+0 and -0), which
makes testing for a zero result more difficult.  Also, addition and
subtraction of two's complement signed numbers are done exactly the
same as addition and subtraction of unsigned numbers (with wraparound on
overflow), which means a CPU can use the same hardware and machine
instructions for both.
```

Grading

Each assignment will be graded in two parts:

1. Does this work indicate any effort? (e.g. it's not copied from a homework for another class or from the book)
2. Three peers will provide short, constructive feedback.

Due Date

This assignment is due on March 22nd, by 11:59 PM Pittsburgh time (currently UTC−4). Remember to convert this time to the timezone you currently reside in.

Question 1

What is internal fragmentation? What is external fragmentation? Compare and contrast the fragmentation present in implicit and explicit lists. Give an example of metadata for an implicit list implementation and a different example for an explicit list implementation and explain the purpose of each.

For a given block, internal fragmentation occurs if payload is smaller than block size (through either padding and/or metadata storage). External fragmentation occurs when there is enough aggregate heap memory, but no single free block is large enough. For an implicit list, metadata includes: size and alloc bit. An example purpose (for alloc bit) is the need to know if a block is free to use for consumption. For an explicit list, metadata includes: size, alloc bit, header, next pointer, prev pointer, footer, mini-block bit, prev-alloc bit. An example purpose (for next pointer) is the need to know where the next free block is while traversing for a malloc call.

NOTE: Depending on when the student completes this assignment, they may or may not know about all the pieces of metadata listed. However, that has no impact on the assignment.

Question 2

For the following two code snippets, assume that allocated blocks must be 16-byte aligned and are implemented with an 8-byte header. Ignore the prologue and the epilogue for now. Which one has more internal fragmentation, and which one has more external fragmentation? Why?

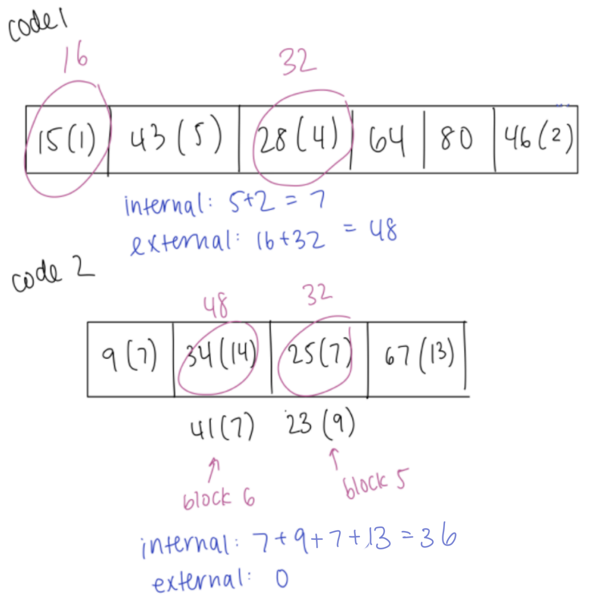
Code 1:

```
block *block1 = malloc(7);
block *block2 = malloc(35);
block *block3 = malloc(20);
block *block4 = malloc(56);
free(block3);
block *block5 = malloc(72);
free(block1);
block *block6 = malloc(38);
```

Code 2:

```
block *block1 = malloc(1);
block *block2 = malloc(26);
block *block3 = malloc(17);
block *block4 = malloc(59);
free(block3);
block *block5 = malloc(15);
free(block2);
block *block6 = malloc(33);
```

The following diagram illustrates what happens when you allocate these blocks. A red circle indicates a free.



In general, the internal fragmentation would include metadata, but since the metadata is the same for both code snippets, we can focus on the padding. As seen from the drawing, the first example clearly has very little internal fragmentation, as the payload takes it very close to the block size, but it has a lot of external fragmentation since we have 48 bytes to spare but can't put the 48 byte block in any open space. The second example has much more internal fragmentation, with payload padding up to 13 or 14 bytes, but there is less external fragmentation because blocks 5 and 6 can fit into the newly freed spaces, leaving no open spaces between blocks.