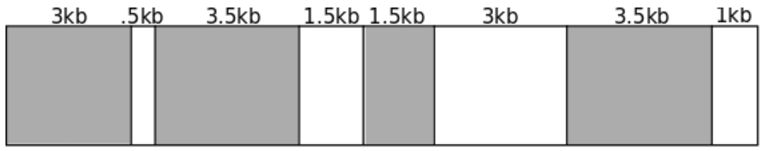
CS241 - Memory

This week we are going to be talking about memory allocators and pointer arithmetic with lots of drawings!

Placement Strategies

How would these three allocation strategies allocate 1KB of memory with the current memory configuration?

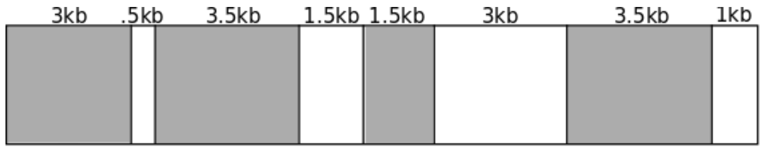
Best Fit



What are some drawbacks to allocating here? (think realloc). What are some drawbacks in general?

-

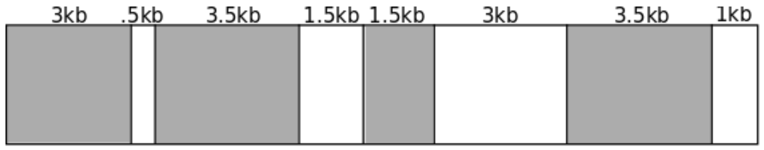
Worst Fit



What are some drawbacks to allocating here? In general?

-

First Fit



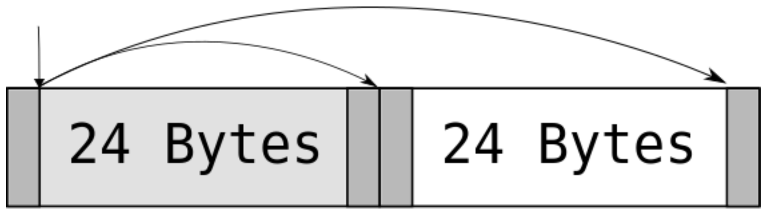
What are some drawbacks to allocating here? In general?

-

Malloc Step by Step

Example: Splitting a Block in Half

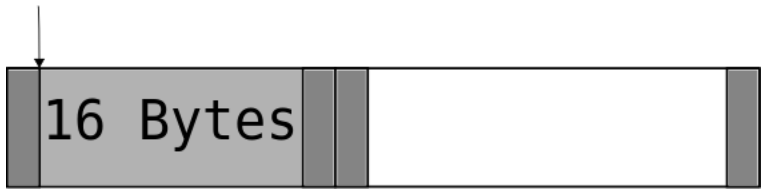
Assume that we have a 64 Byte Heap with 4 Byte tags. If we have an empty heap and a starting pointer, what does it look like after malloc(24)? What pointer atihmetic do we have to do? (assume no rounding and no free list).



After subtracting the bytes for the original metadata tags (a total of 8 bytes), we have 56 bytes of space left to work with. If we need to serve an allocation request of 24, we are going to do the following jumps

```
if(block->size < needed)
    //Next!
else if(block->size < needed + 2 * sizeof(metdata))
    //Give them the entire block
else
    new_block = start + needed + 2 * sizeof(metdata)
    // And write the meta data blocks
```

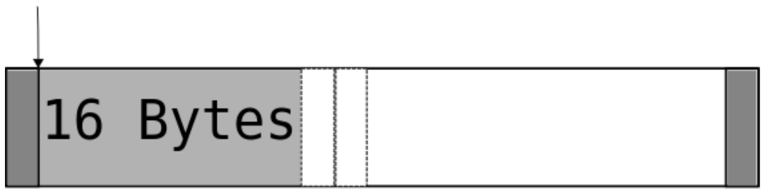
malloc(16), 64 Byte Heap, 4B Boundary Tags



How would I satisfy this allocation request?

-

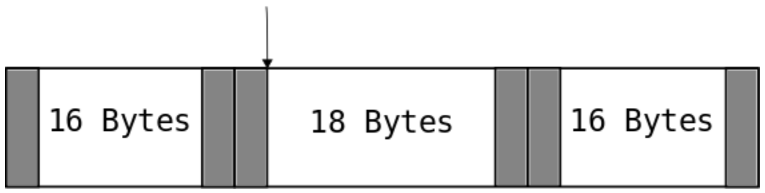
realloc(arrow, 24)), 64 Byte Heap, 4B Boundary Tags



How would I satisfy this allocation request?

-

free(arrow), 64 Byte Heap, 4B Boundary Tags



How would I satisfy this allocation request? What about double coalescing? What about having a free list?

-