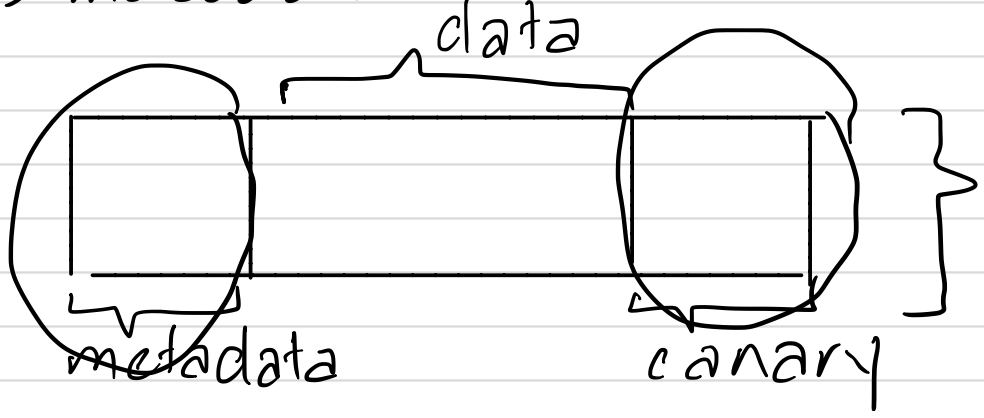
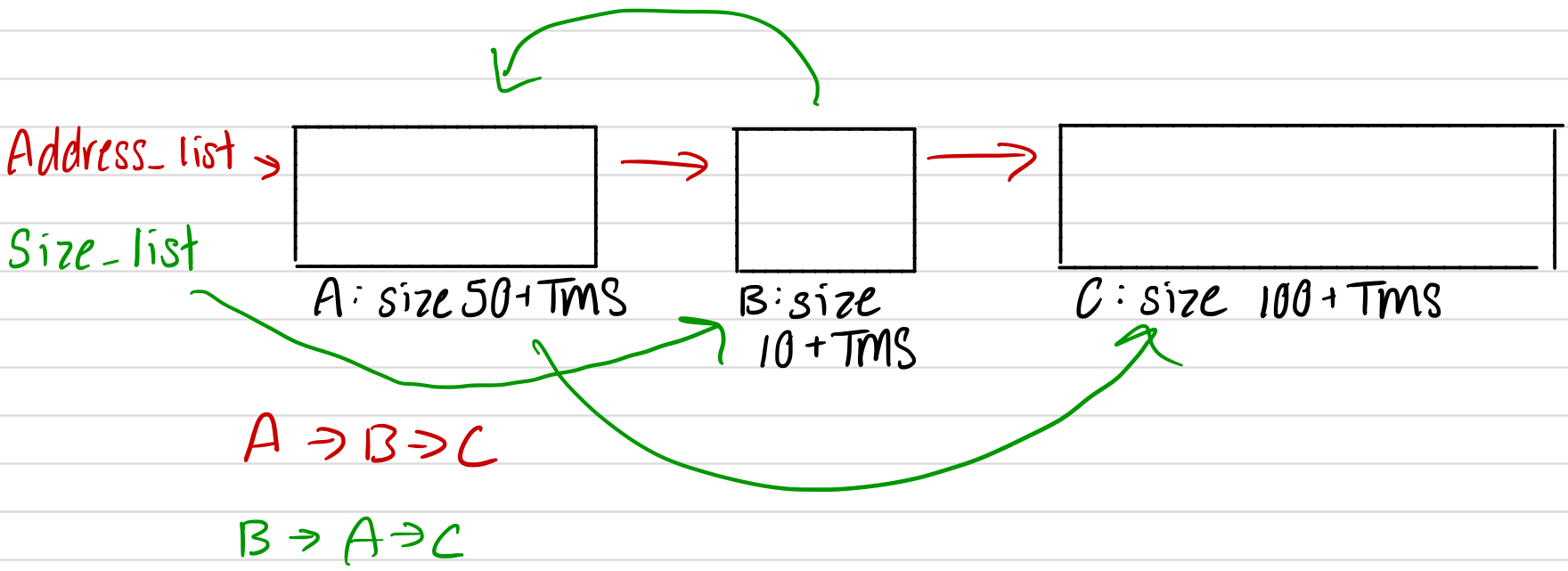
Malloc Recitation

Part A: The Block

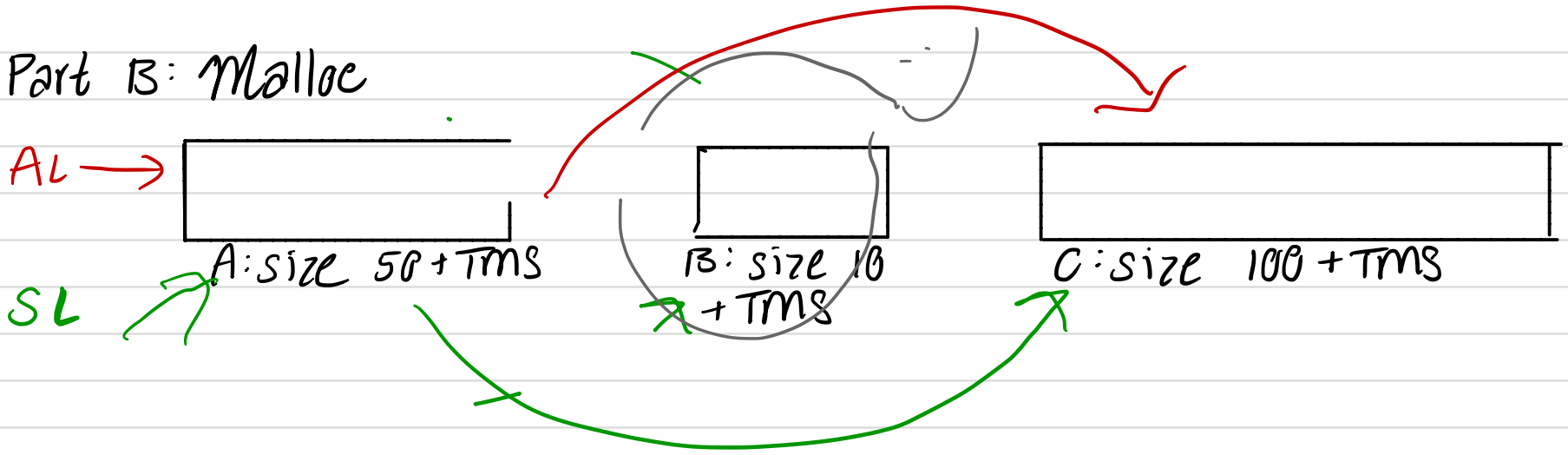
```
typedef struct metadata {
        unsigned long canary,
        unsigned long size;
        struct metadata * next_addr,
        struct metadata * next_size,
        struct metadata * prev_addr,
        struct metadata* prev_size,
} metadata_t*
```

user req size + TMS





Part B: Malloc



Case 1: my_malloc(10)

① Add TMS to user requested to get size of block actually want
10 + TMS

② Iterate through size list to find block of best size

③ Once we find our perf block remove from address + size lists

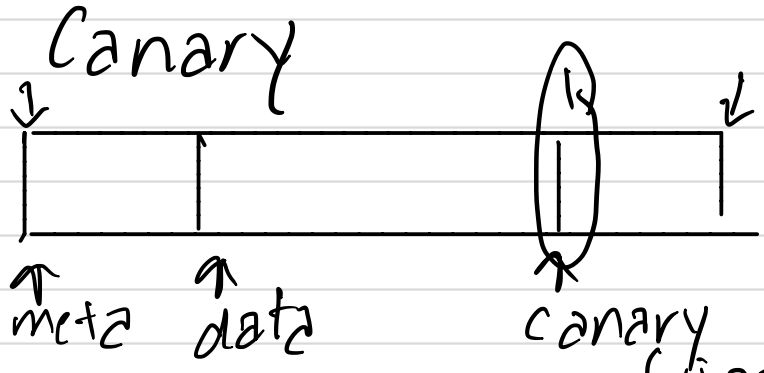
B ⇒ pa ⇒ na = B ⇒ na

B ⇒ na ⇒ pa = B ⇒ pa

B ⇒ ps ⇒ ns = B ⇒ ns

B ⇒ ns ⇒ ps = B ⇒ ps

④ Set Canary

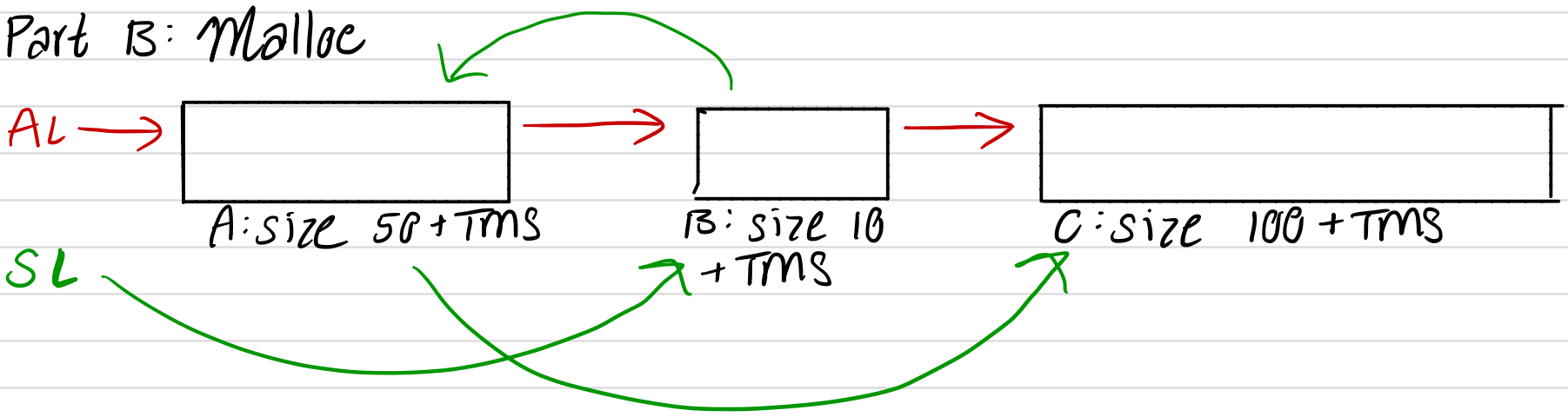


B ⇒ canary = lovely function

(uint8_t*)B + B ⇒ size - sizeof(unsigned long

⑤ Return pointer to user data
return B+1

Part B: Malloc



Case 2: my_malloc(40)

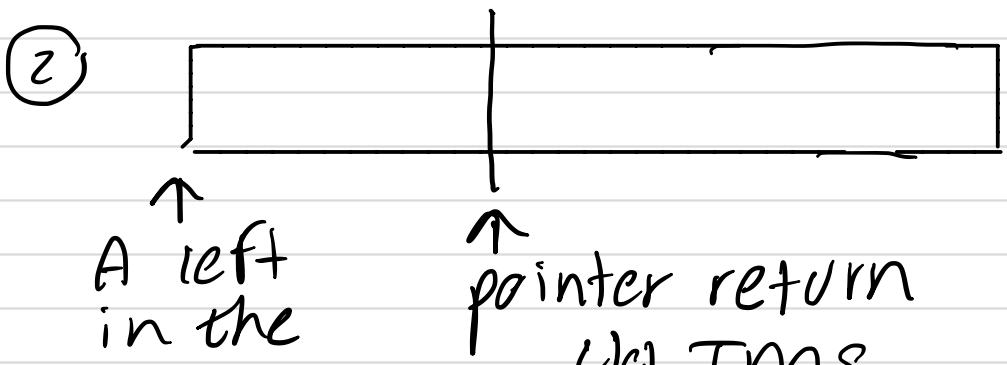
Note: consider min_block_size = 5

① 40 + TMS

split block (!)

40 + TMS + MIN_BLOCK_SIZE

(1 + TMS)

② 

→ ③a set canaries

③b proper spot in address list!

④a return a pointer to the user data

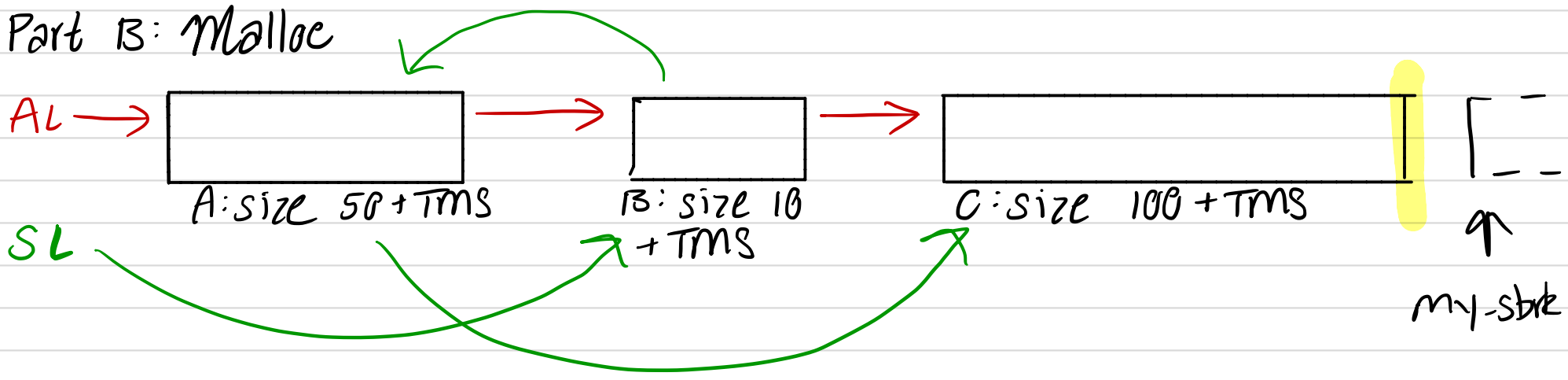
④b remove size list

⑤a null out all pointers

④c A→size = A→size - 40 + Tms

④d Reenter into size list

Part B: Malloc



Case 3: my_malloc(97)
Note: consider min_block_size = 5

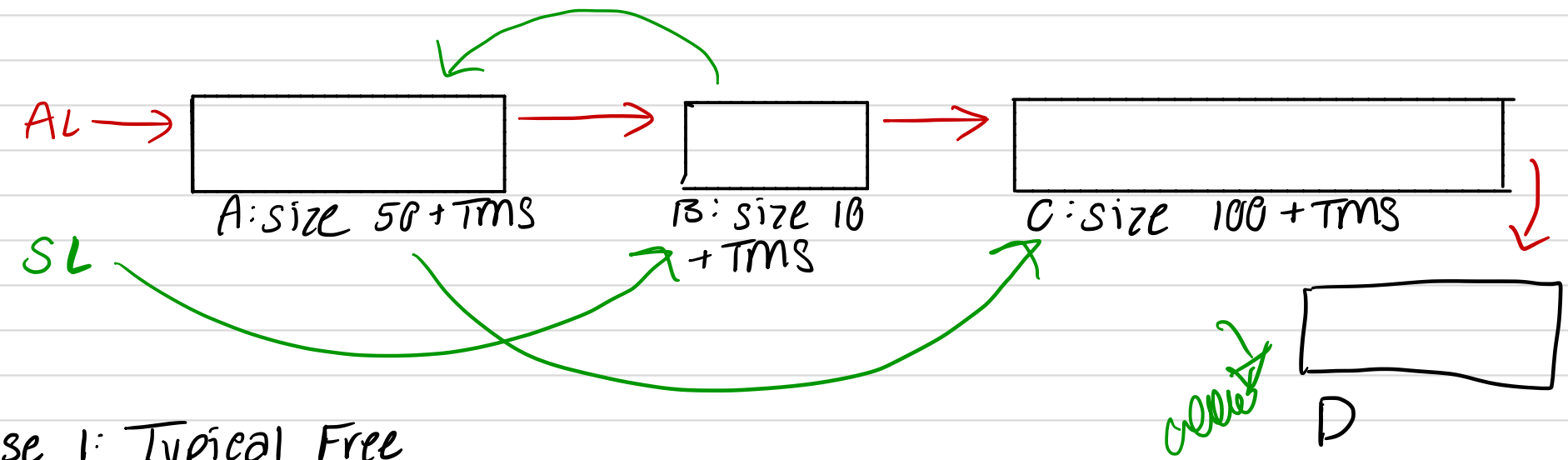
① 97 + TMS
97 + TMS + MBS(5) > C->size

ONLY WAY TO SPLIT
C->size > user_req + TMS + MBS

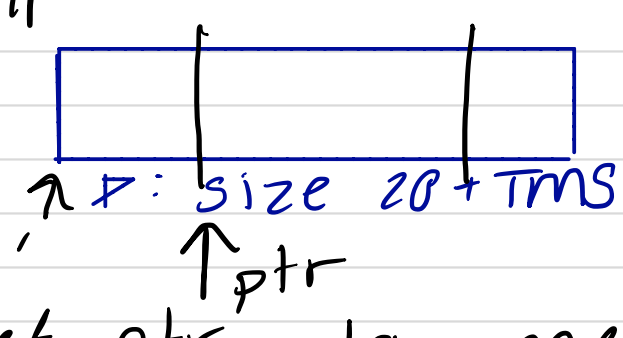
② Call my_sbrk!

③ Re-go through the entire size-list to find block we want to return to user

Part C: Free



Case 1: Typical Free



↑ptr

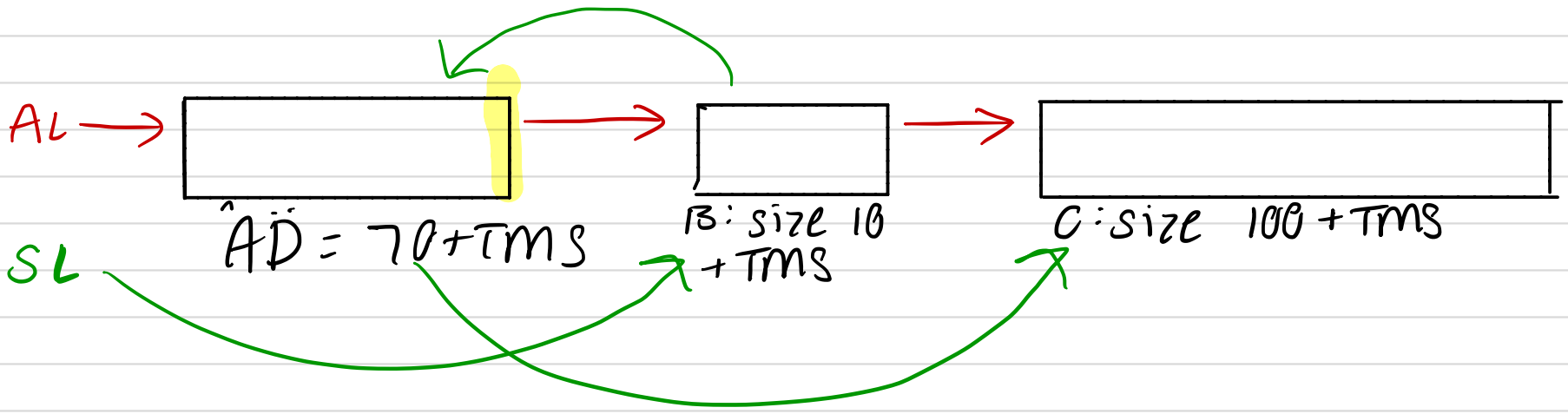
① get ptr to metadata of block
(metadata_t*)ptr - 1

② check canaries
↳ if bad
set CANARY_CORRUPTED & return

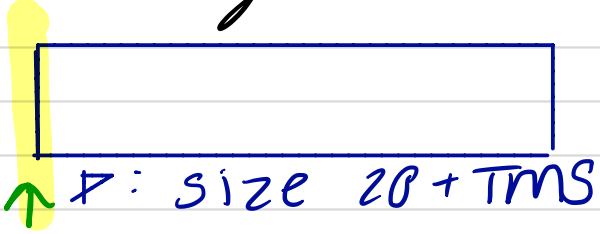
③ iterate through addr list to find proper place of our block
(uintptr_t*)
↑ allows us to compare pointers!

④ Add block to size list if no merging

Part C: Free



Case 2: Left Merge



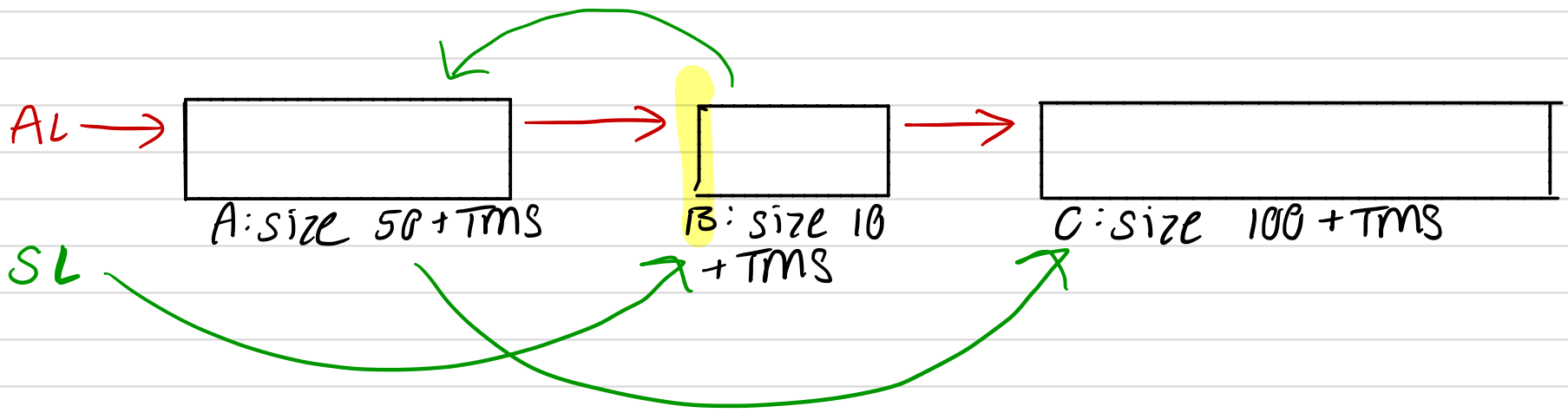
A + A ⇒ size = D

① Keep A in the address list where it is

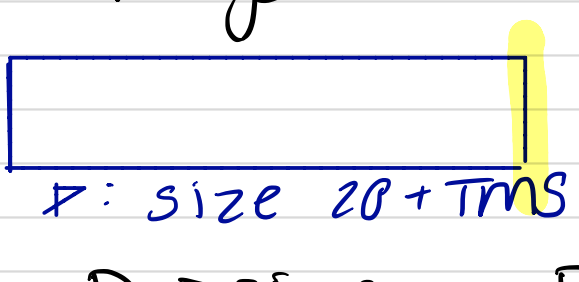
② A⇒size = A⇒size + D⇒size

③ Remove A from size list & reenter it in proper location

Part C: Free



Case 3: Right Merge



D + D → size = B

① D → pa = B → pa
D → na = B → na
B → pa → na = D
B → na → pa = D
D → size = B → size + D → size
Remove B from size list
Add DB into size list

Part D: Realloc

my_realloc(ptr, size)

Case 1: Pointer null

ret my_malloc(size)

Case 2: Pointer non-null, canaries corrupted

errno = CANARY_CORRUPTED

return null

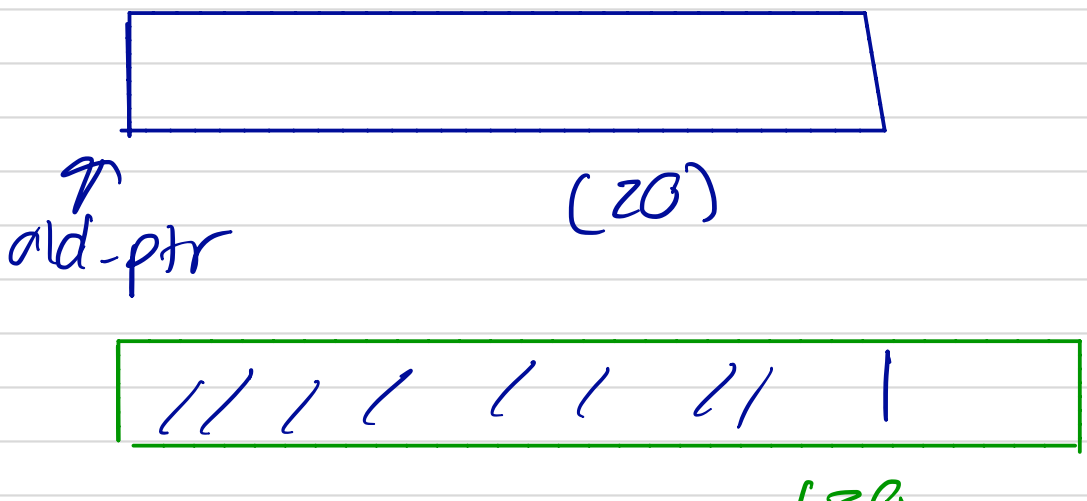
Case 3: Size == 0, Pointer non-null

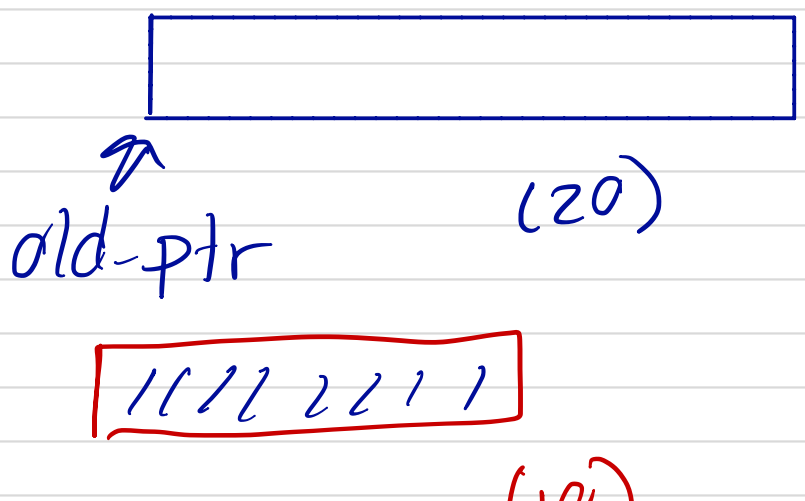
free(ptr)

return null

Part D: Realloc
Case 4: Pointer non-null & size != 0

new_ptr = my_malloc(size)



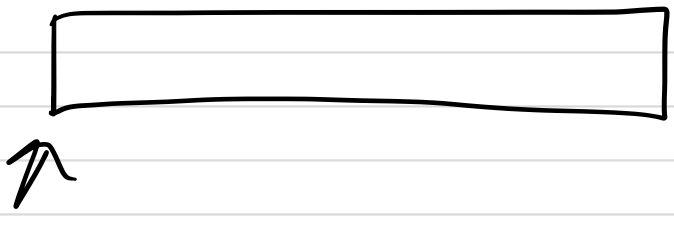


Highly suggest checking out memcpy

my_free(old_ptr)
return new_ptr

Part E: Calloc

my-malloc



Set everything in the user portion of block to 0

highly suggest checking out memset