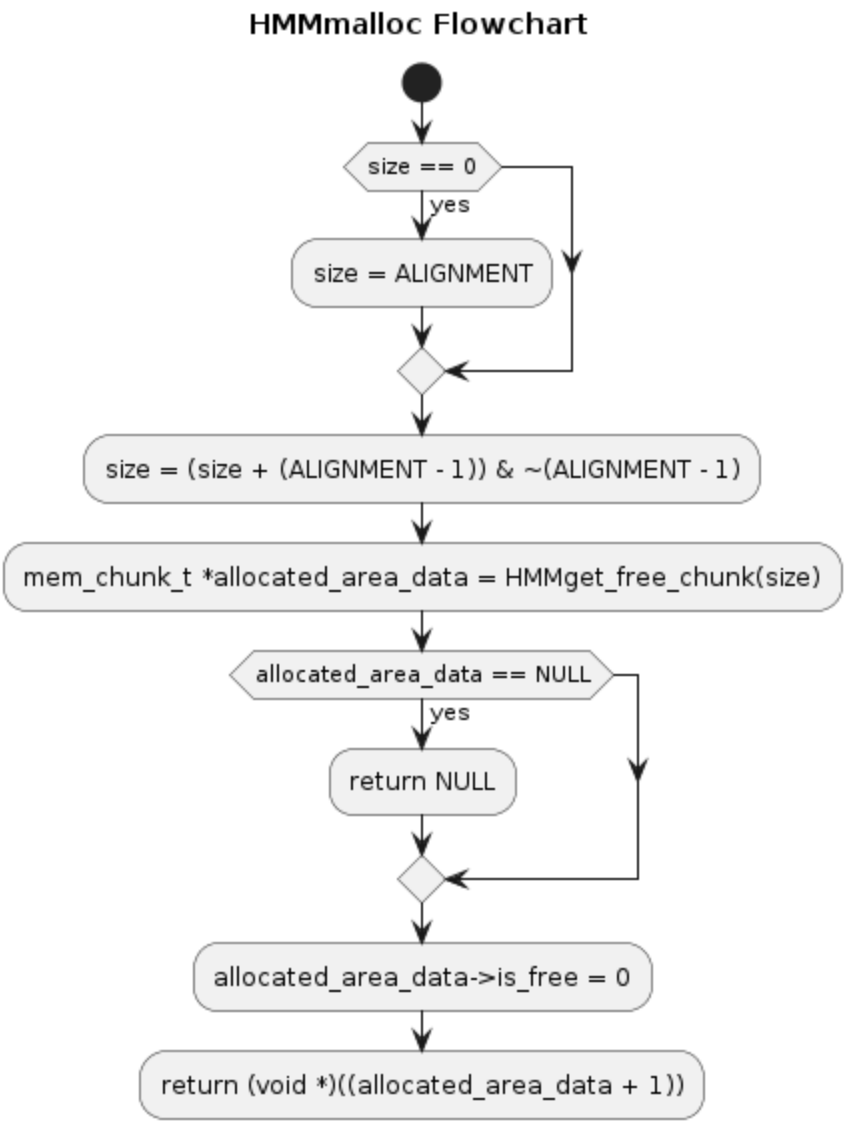
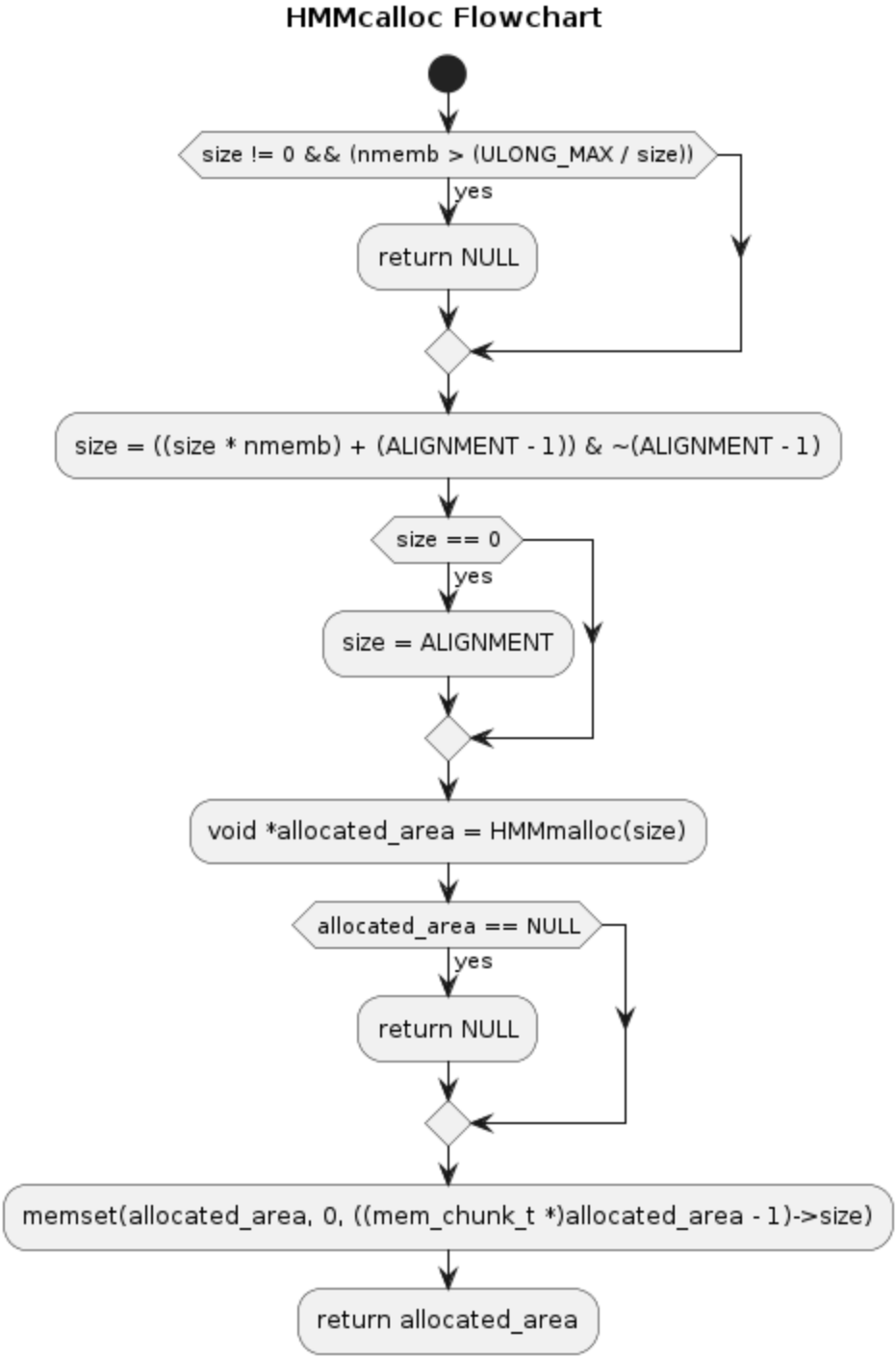
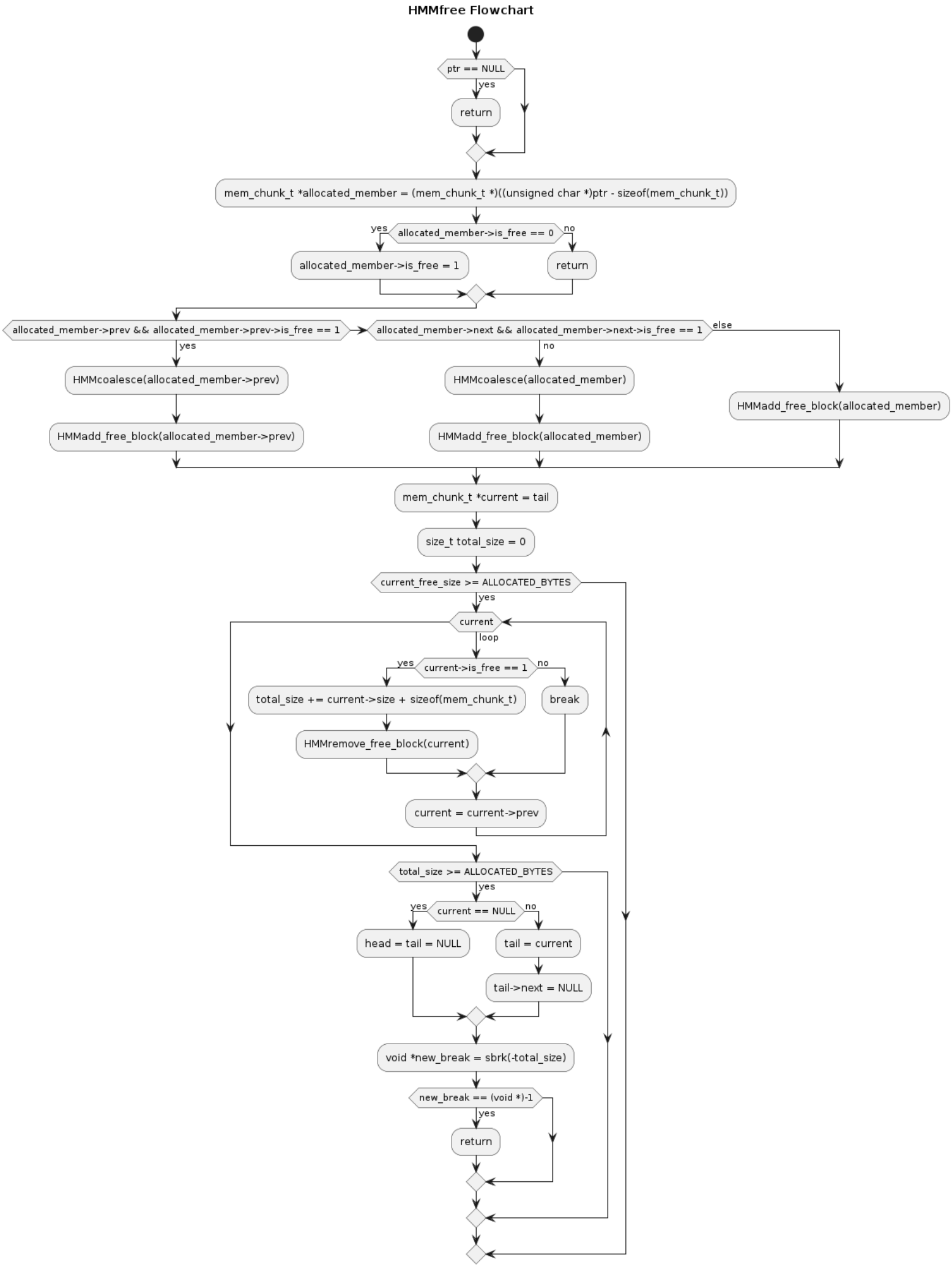
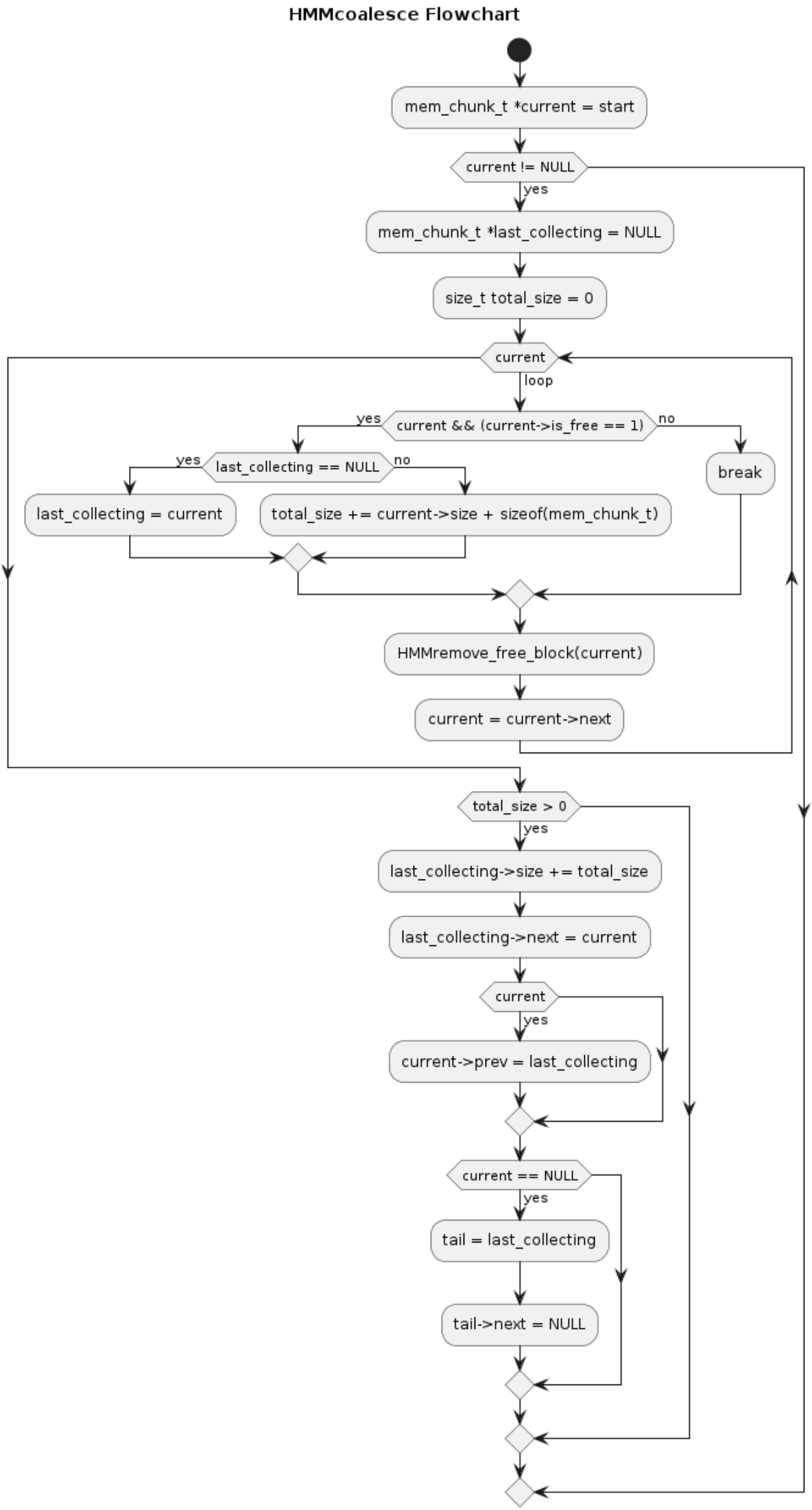
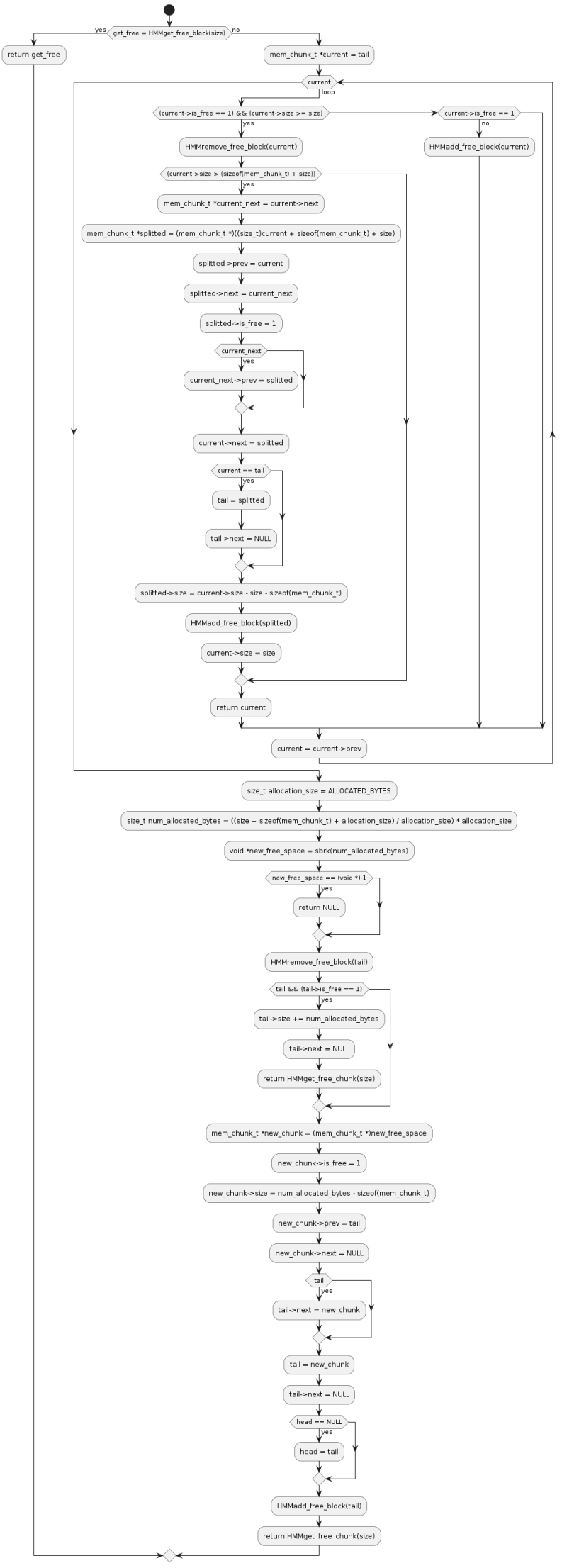

HMMmalloc Flowchart
size == 0
yes
size = ALIGNMENT
size = (size + (ALIGNMENT - 1)) & ~(ALIGNMENT - 1)
mem_chunk_t *allocated_area_data = HMMget_free_chunk(size)
allocated_area_data == NULL
yes
return NULL
allocated_area_data->is_free = 0
return (void *)((allocated_area_data + 1))

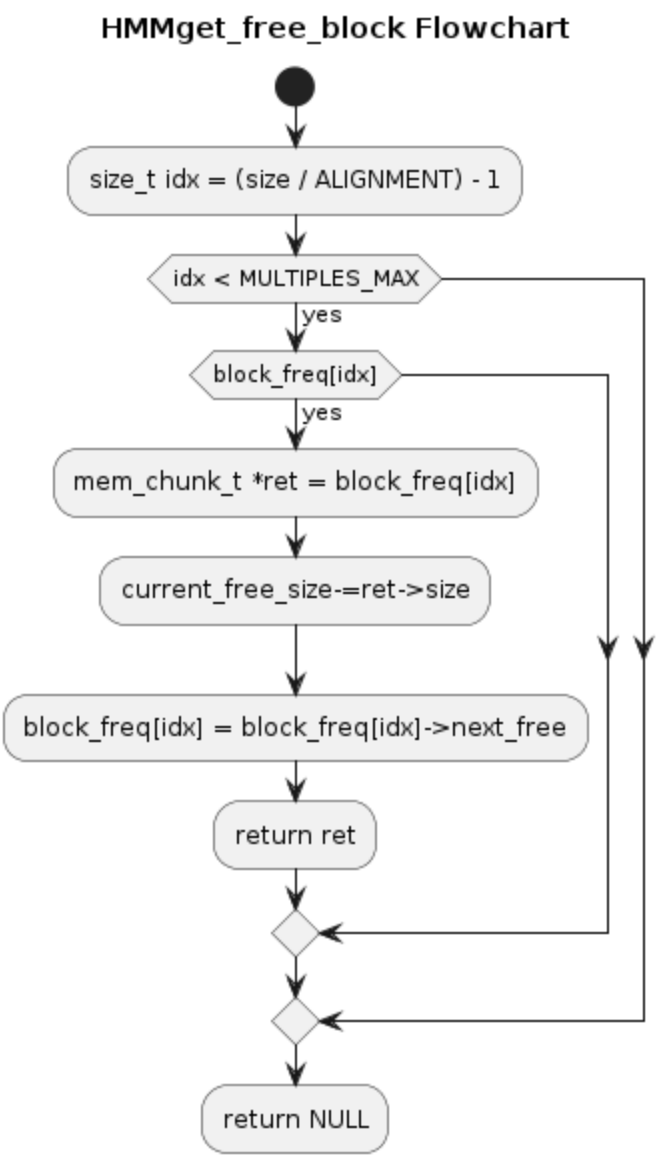

HMMcalloc Flowchart
size != 0 && (nmemb > (ULONG_MAX / size))
yes
return NULL
size = ((size * nmemb) + (ALIGNMENT - 1)) & ~(ALIGNMENT - 1)
size == 0
yes
size = ALIGNMENT
void *allocated_area = HMMmalloc(size)
allocated_area == NULL
yes
return NULL
memset(allocated_area, 0, ((mem_chunk_t *)allocated_area - 1)->size)
return allocated_area

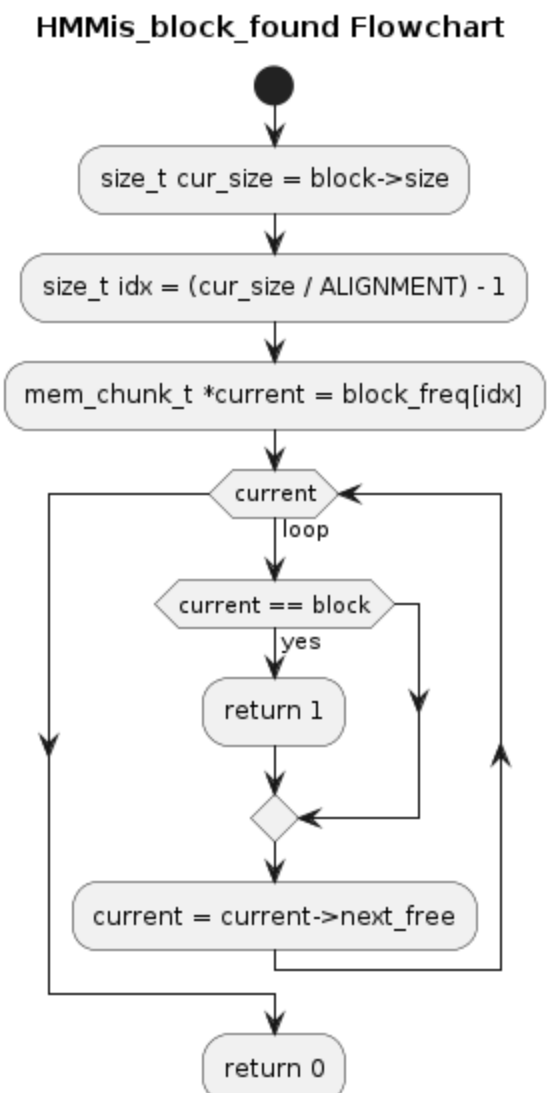
HMMrealloc Flowchart
size = ((size) + (ALIGNMENT - 1)) & ~(ALIGNMENT - 1)
ptr == NULL
yes
return HMMmalloc(size)
size == 0
yes
HMMfree(ptr)
return HMMmalloc(ALIGNMENT)
size == ((mem_chunk_t *)ptr - 1)->size
yes
return ptr
void *allocated_chunk = HMMmalloc(size)
allocated_chunk == NULL
yes
return NULL
size_t allocation_size
yes
((mem_chunk_t *)ptr - 1)->size < ((mem_chunk_t *)allocated_chunk - 1)->size
no
allocation_size = ((mem_chunk_t *)ptr - 1)->size
allocation_size = ((mem_chunk_t *)allocated_chunk - 1)->size
memcpy(allocated_chunk, ptr, allocation_size)
HMMfree(ptr)
return (void *)allocated_chunk


HMMfree Flowchart
ptr == NULL
yes
return
mem_chunk_t *allocated_member = (mem_chunk_t *)((unsigned char *)ptr - sizeof(mem_chunk_t))
yes
allocated_member->is_free == 0
no
allocated_member->is_free = 1
return
allocated_member->prev && allocated_member->prev->is_free == 1
allocated_member->next && allocated_member->next->is_free == 1
else
yes
no
HMMcoalesce(allocated_member->prev)
HMMcoalesce(allocated_member)
HMMadd_free_block(allocated_member)
HMMadd_free_block(allocated_member->prev)
HMMadd_free_block(allocated_member)
mem_chunk_t *current = tail
size_t total_size = 0
current_free_size >= ALLOCATED_BYTES
yes
current
loop
yes
current->is_free == 1
no
total_size += current->size + sizeof(mem_chunk_t)
break
HMMremove_free_block(current)
current = current->prev
total_size >= ALLOCATED_BYTES
yes
yes
current == NULL
no
head = tail = NULL
tail = current
tail->next = NULL
void *new_break = sbrk(-total_size)
new_break == (void *)-1
yes
return

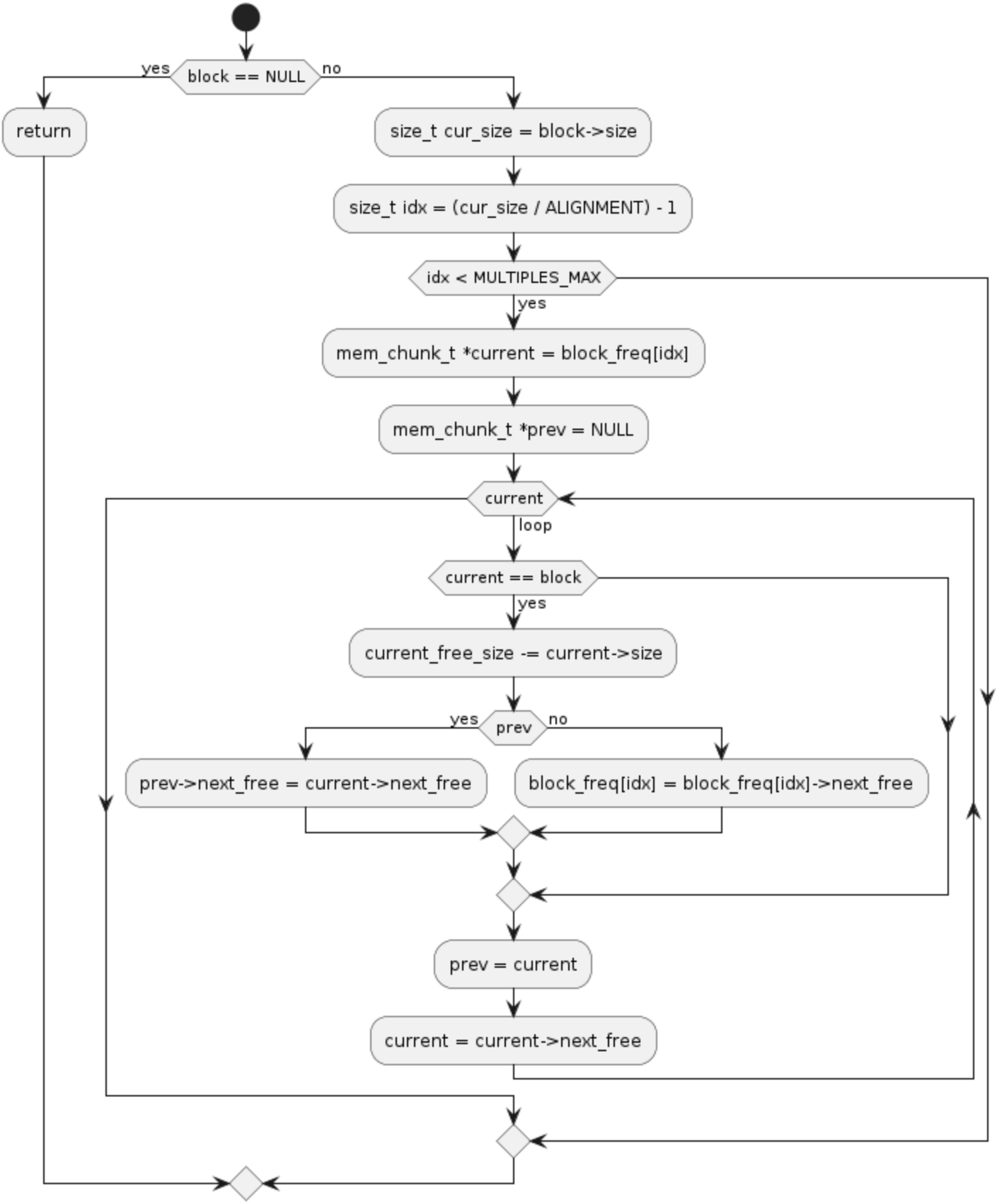
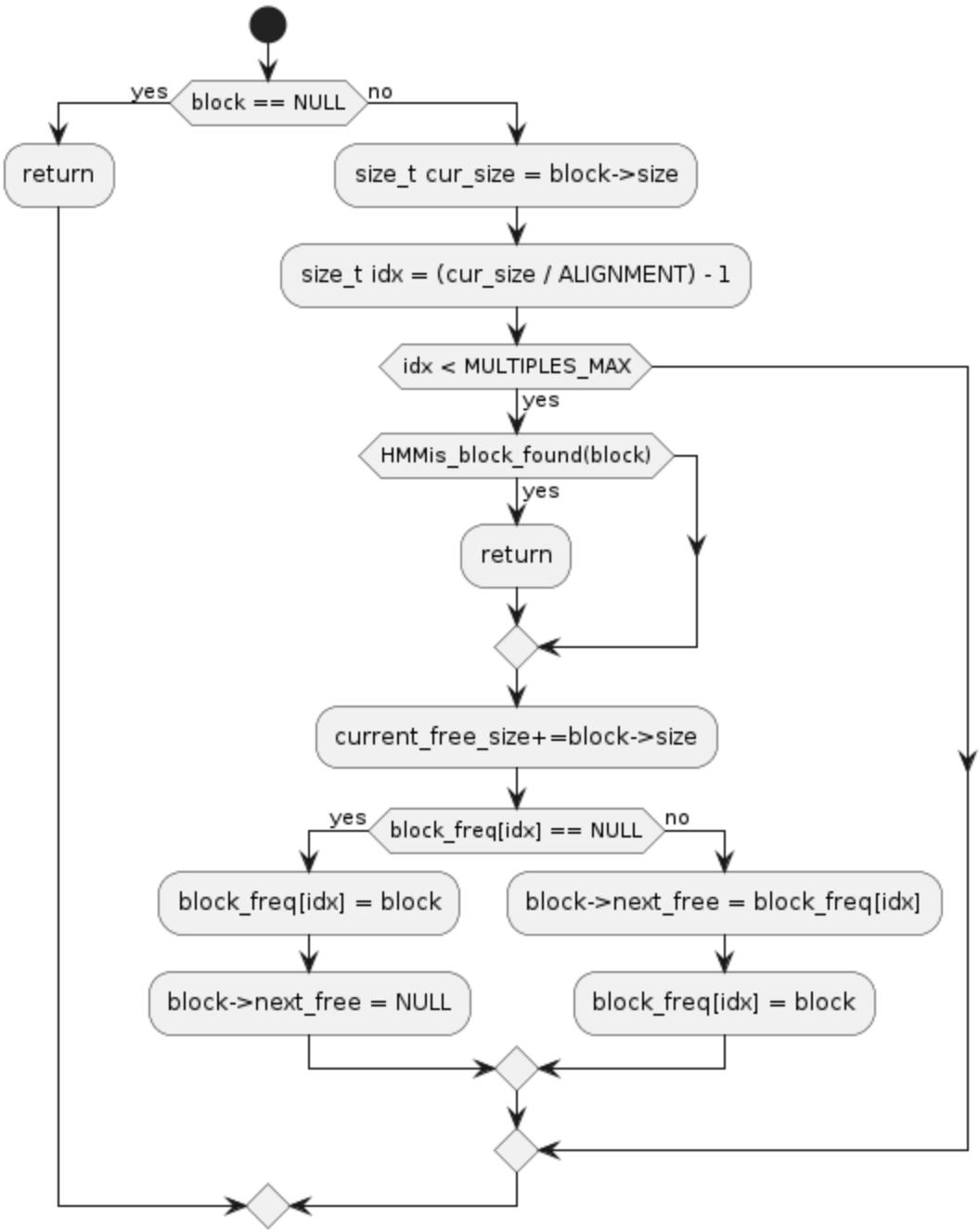

HMMcoalesce Flowchart
mem_chunk_t *current = start
current != NULL
yes
mem_chunk_t *last_collecting = NULL
size_t total_size = 0
current
loop
yes
current && (current->is_free == 1)
no
break
yes
last_collecting == NULL
no
last_collecting = current
total_size += current->size + sizeof(mem_chunk_t)
HMMremove_free_block(current)
current = current->next
total_size > 0
yes
last_collecting->size += total_size
last_collecting->next = current
current
yes
current->prev = last_collecting
current == NULL
yes
tail = last_collecting
tail->next = NULL


HMMget_free_chunk Flowchart
get_free = HMMget_free_block(size)
yes
no
return get_free
mem_chunk_t *current = tail
current
loop
(current->is_free == 1) && (current->size >= size)
current->is_free == 1
yes
no
HMMremove_free_block(current)
HMMadd_free_block(current)
(current->size > (sizeof(mem_chunk_t) + size))
yes
mem_chunk_t *current_next = current->next
mem_chunk_t *splitted = (mem_chunk_t *)((size_t)current + sizeof(mem_chunk_t) + size)
splitted->prev = current
splitted->next = current_next
splitted->is_free = 1
current_next
yes
current_next->prev = splitted
current->next = splitted
current == tail
yes
tail = splitted
tail->next = NULL
splitted->size = current->size - size - sizeof(mem_chunk_t)
HMMadd_free_block(splitted)
current->size = size
return current
current = current->prev
size_t allocation_size = ALLOCATED_BYTES
size_t num_allocated_bytes = ((size + sizeof(mem_chunk_t) + allocation_size) / allocation_size) * allocation_size
void *new_free_space = sbrk(num_allocated_bytes)
new_free_space == (void *)-1
yes
return NULL
HMMremove_free_block(tail)
tail && (tail->is_free == 1)
yes
tail->size += num_allocated_bytes
tail->next = NULL
return HMMget_free_chunk(size)
mem_chunk_t *new_chunk = (mem_chunk_t *)new_free_space
new_chunk->is_free = 1
new_chunk->size = num_allocated_bytes - sizeof(mem_chunk_t)
new_chunk->prev = tail
new_chunk->next = NULL
tail
yes
tail->next = new_chunk
tail = new_chunk
tail->next = NULL
head == NULL
yes
head = tail
HMMadd_free_block(tail)
return HMMget_free_chunk(size)


HMMget_free_block Flowchart
size_t idx = (size / ALIGNMENT) - 1
idx < MULTIPLES_MAX
yes
block_freq[idx]
yes
mem_chunk_t *ret = block_freq[idx]
current_free_size-=ret->size
block_freq[idx] = block_freq[idx]->next_free
return ret
return NULL


HMMis_block_found Flowchart
size_t cur_size = block->size
size_t idx = (cur_size / ALIGNMENT) - 1
mem_chunk_t *current = block_freq[idx]
current
loop
current == block
yes
return 1
current = current->next_free
return 0

HMMremove_free_block Flowchart



HMMadd_free_block Flowchart