

Midterm – spring 2020

Memory Management

It's your first day at your Google internship. During your interview you mentioned that you studied various methods for memory management, and among them was one that used a boundary tag approach. You mentioned that in class you worked through the data structures needed and the pseudocode implementations of the algorithms allocate and free. You also described the Safearray class.

So ... for your first assignment your mentor at Google asked you to rework the one-dimensional Safearray template class so that it allocates memory by including a "block" object where blocks are object allocated from a memory pool that is maintained in a **boundary tag implementation** (doubly linked, size and tag fields etc.). "Blocks" are those tagged memory regions that you described to her during the interview that are either on a linked list of available blocks or currently allocated for use.

Your mentor says the block class will define its own versions of allocate and free and will allocate blocks from the memory pool that it controls. The block class will allocate a large area of memory when the first block is created and allocate from that memory pool from then on anytime a new block is required. The block class will get the initial memory by calling the system new.

What needs to be done:

1. Define the block class with appropriate functionality, memory management (new and delete etc.,) and public member functions so that it might be used by a one-dimensional Safearray template class that looks something like this;

```
class SA<t>{  
    int low, hi;  
    block(low,high,sizeof(t)) b;  
public: ...
```

2. Write a test simulation along the lines of:

```
for(int i=0;i<100; i++){  
    block b(random value for low, random value for high, sizeof(int));  
}
```

where the modified constructors and destructors report statistics such as number of blocks on the free list, average block size on list, number of blocks to probe till a suitable block is found to allocate from, etc. the kind of information you would imagine required if you were to do this as a programmer for Google.

3. Write a Safearray class that uses a block to get memory and show by examples that this Safearray class functions identically to the original one where memory was obtained directly from the default allocator (system new and delete).

After you think about the problem for a bit you turn to your mentor at Google and say (gently) that you feel the problem is perhaps, ... a bit underspecified ... like, "what are the member functions, how do I return a pointer to appropriate beginning of a granted block ...???" You mentor smiles at you and says "we trust you to make your own decisions We hired you after all. Let's see some good design decisions within the confines of the spec, also don't overlook: clear and efficient code and, of course, useful documentation. ...". At this point you are not so sure any more about this internship, but you are in Mountain View California, and can't yet afford a ticket home ... so ...

Note: You might look at the material about placement new for the block class:

<https://venus.cs.qc.cuny.edu/~waxman/780/Placement%20New%20from%20common%20knowledge.pdf> because you will need to construct the memory in a block to be a t object.