

Notice for bonus:

- Completed the recursive memory for part II and III
- Using own P3 implementation (page table)
- Completed by the original due date

### Report (VMPool):

Overview:

VMPool uses an array for the list of region descriptors. It can store at max 512 region descriptors and uses the first 4K frame at the specified base address to do this. The array maintains sorting by region base addresses in ascending order. Any time a region is released, then the array removes that element from the list, and all regions after are shifted left. Any time a region is allocated, all regions are shifted right to allow the insertion. The allocate function uses the first fit algorithm for allocating memory and can find large enough regions easily because the region base addresses are in order.

Constructor(\_base\_address, \_size, \_frame\_pool, \_page\_table)

1. Sets member variables for base\_address, size, page\_table, and frame\_pool
2. It registers the pool using the "this pointer" to the page\_table
3. Sets up the region array by setting it to the base\_address of the VMPool, and then sets the first descriptor to describe the regions list itself. So the first region has base address the same as the base address of the VMPool, and a size of 1 Frame or 4KB.
4. Iterates through all 511 remain region descriptors and sets their base addresses and sizes to 0

Allocate(\_size)

1. Convert the size in bytes to the number of frames needed for that number of bytes
2. Iterate through regions list until a large enough gap between two consecutive regions is found or the end of the regions list is reached
3. If a large enough gap is found then we need to insert a region in between the two consecutive regions by shifting right all regions including and after the location we want to insert the new region at.
4. We create a new region at that location with the base address directly after the last region's last address and the size as specified.
5. Return the new regions' base address

Release(\_start\_address)

1. Iterate through all regions to find the region that needs to be released
2. We release the region by iterate through all frames in that region and calling page\_table->free\_page with the frame number.
3. We shift the regions left because that region is now released.

is\_legitimate(address)

1. By default, if the address is the same as the VMPool base address then we return true because that address will always be used to manage the regions list
2. We iterate through all of the regions and if the address is within a region then return true
3. Otherwise, return false