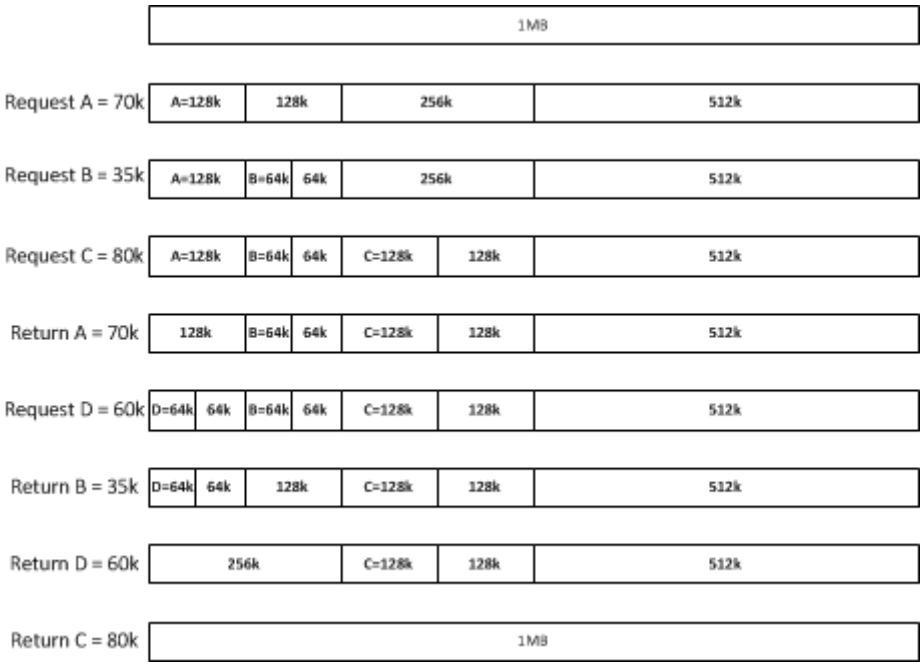
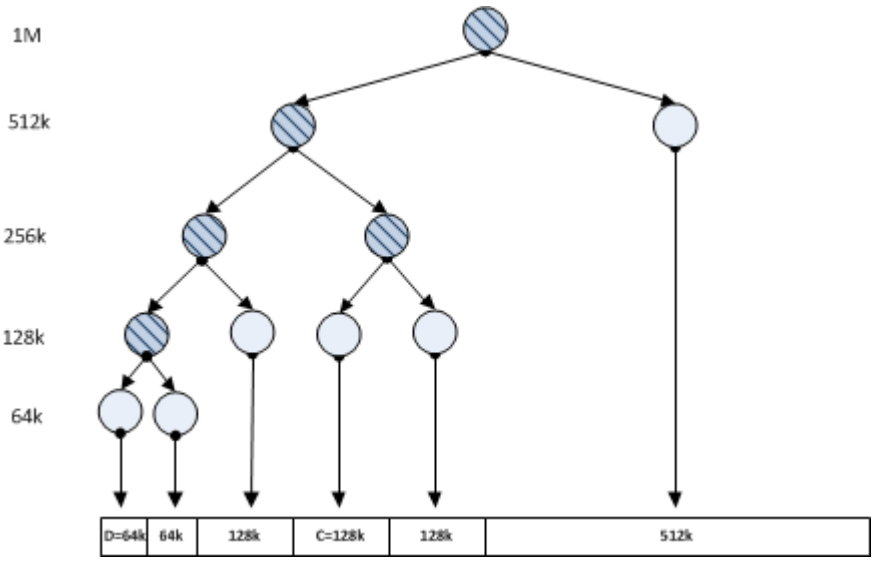
Homework 2
Seth Denney

7.6
a) The largest that the swapped-out process could have been is 8MB, as that is the largest contiguous block of currently-available memory.
b) 1MB + 2MB + 5MB → 8MB
c) 3MB process to be brought into memory...
BEST – It will take the 3MB block near the end.
FIRST – It will take part of the 4MB block near the front.
NEXT – It will take part of the 5MB block immediately following X.
WORST – It will take part of the 8MB block in the middle (the largest).

7.7
a)



b)



7.12
a) 2^{16} pages : 2^{10} byte page size → 26 bits in a logical address (16 for page number, 10 for offset)
b) 2^{10} byte page size → 2^{10} byte frame size
c) 2^{32} bytes of physical memory & 2^{10} byte frame size → 2^{22} frames in physical memory → 22 bits in physical address for frame number
d) 2^{16} pages of logical address space → 2^{16} entries in page table
e) 22 bits for frame number *(from c)* + valid/invalid bit → 23 bits in each page table entry

8.1
a) In general, a virtual memory address will contain some number of page references (depending on the structure of the page tables, tiered or single-table) followed by an offset. The page(s) are located in the page table(s) from a combination of the page number(s) from the virtual address and a page table initial offset (usually a register). This will produce the appropriate physical frame number, which, when combined with the offset from the virtual address, will produce a physical memory address.
b)
i) virtual address: 1052 → page: 1 offset: 052 → frame: 7 → physical address: 7052
ii) virtual address: 2221 → page:2 → invalid address
iii) virtual address: 5499 → page: 5 offset: 499 → frame: 0 → physical address: 0499

8.3
a) 4 MB
b) each PTE contains 32 bits (4 bytes) of info & there are 2^6 frames/PTEs available → 256 bytes
**an overflow does not use additional memory, as the chain pointer stored in the PTE simply points to another entry in the extant table, whose size is set at 2^6 entries

8.4 ***NOTE: On Clock: Shaded = Use Bit Set; * = pointer*

	7	0	1	2	0	3	0	4	2	3	0	3	2
FIFO #PF: 7 Miss: 70%	7	7 0	7 0 1	2 0 1	2 0 1	2 3 1	2 3 0	4 3 0	4 2 0	4 2 3	0 2 3	0 2 3	0 2 3
LRU #PF: 7 Miss: 70%	7	7 0	7 0 1	2 0 1	2 0 1	2 3 1	2 3 0	4 3 0	4 2 0	4 2 3	0 2 3	0 2 3	0 2 3
Clock #PF: 6 Miss: 60%	7 *	7 0 *	7* 0 1	2 0* 1	2 0* 1	2* 0 3	2* 0 3	4 0* 3	4 2 3*	4 2 3*	4* 2 0	3 2* 0	0 2* 3
Optimal #PF: 4 Miss: 40%	7	7 0	7 0 1	2 0 1	2 0 1	2 0 3	2 0 3	2 4 3	2 4 3	2 4 3	2 0 3	2 0 3	2 0 3

8.10
Page Size: 2^{12} bytes
PTE Size: 2^2 bytes
Memory: 2^{64} bytes
Level 7: 2^{64} bytes → 2^{52} pages, each mapped to 4 byte PTE in Level 6...
Level 6: 2^{54} bytes → 2^{42} pages, each mapped to 4 byte PTE in Level 5...
Level 5: 2^{44} bytes → 2^{32} pages, each mapped to 4 byte PTE in Level 4...
Level 4: 2^{34} bytes → 2^{22} pages, each mapped to 4 byte PTE in Level 3...
Level 3: 2^{24} bytes → 2^{12} pages, each mapped to 4 byte PTE in Level 2...
Level 2: 2^{14} bytes → 2^2 pages, each mapped to 4 byte PTE in Level 1...
Level 1: 2^4 bytes → 1 page

8.16

A simple mechanism to include this capability could involve decrementing Q when L is reached without a page fault occurring, and conversely incrementing Q when M is reached if Q page faults have occurred before M is reached. This would leave Q untouched when Q is reached in the expected interval, but would adjust it to be (potentially) more appropriate if the page fault rate is significantly higher or lower than expected. NOTE: The adjustments of Q would occur after any resident size adjustments have been made.