Discussion Section: Week 10 (Solutions)
Problem 1:

Consider a system in which the memory has the following hole sizes in the following memory order:
1KB, 4 KB, 15 KB, 20 KB, 4 KB, 7 KB, 18 KB, 12 KB, 15 KB, 9 KB

You are given successive requests for program segments in the following order:
10 KB, 5 KB, 3 KB, 2 KB, 19 KB, 9 KB, 24 KB, 10 KB.

For each of the following algorithms, show how the holes get filled for each of the above requests. If a particular request cannot be satisfied, you can skip it (but do mention which ones cannot be satisfied):

1. First fit:
   10 KB: 1, 4, **5**, 20, 4, 7, 18, 12, 15, 9
   5 KB: 1, 4, 20, 4, 7, 18, 12, 15, 9
   3 KB: 1, **1**, 20, 4, 7, 18, 12, 15, 9
   2 KB: 1, 1, **18**, 4, 7, 18, 12, 15, 9
   19 KB: Can't fit
   9 KB: 1, 1, **9**, 4, 7, 18, 12, 15, 9
   24 KB: Can't fit
   10 KB: 1, 1, 9, 4, 7, **8**, 12, 15, 9
2. Best fit:
   10 KB: 1, 4, 15, 20, 4, 7, 18, **2**, 15, 9
   5 KB: 1, 4, 15, 20, 4, **2**, 18, 2, 15, 9
   3 KB: 1, **1**, 15, 20, 4, 2, 18, 2, 15, 9
   2 KB: 1, 1, 15, 20, 4, 18, 2, 15, 9
   19 KB: 1, 1, 15, **1**, 4, 18, 2, 15, 9
   9 KB: 1, 1, 15, 1, 4, 18, 2, 15
   24 KB: Can't fit
   10 KB: 1, 1, **5**, 1, 4, 18, 2, 15
3. Worst fit:
   10 KB: 1, 4, 15, **10**, 4, 7, 18, 12, 15, 9
   5 KB: 1, 4, 15, 10, 4, 7, **13**, 12, 15, 9
   3 KB: 1, 4, **12**, 10, 4, 7, 13, 12, 15, 9
   2 KB: 1, 4, 12, 10, 4, 7, 13, 12, **13**, 9
   19 KB: Can't fit
   9 KB: 1, 4, 12, 10, 4, 7, **4**, 12, 13, 9
   24 KB: Can't fit
   10 KB: 1, 4, 12, 10, 4, 7, 4, 12, **3**, 9
4. Next fit:
   10 KB: 1, 4, **5**, 20, 4, 7, 18, 12, 15, 9
   5 KB: 1, 4, 20, 4, 7, 18, 12, 15, 9
   3 KB: 1, 4, **17**, 4, 7, 18, 12, 15, 9
   2 KB: 1, 4, **15**, 4, 7, 18, 12, 15, 9
   19 KB: Can't fit
   9 KB: 1, 4, **6**, 4, 7, 18, 12, 15, 9
   24 KB: Can't fit
   10 KB: 1, 4, 6, 4, 7, **8**, 12, 15, 9

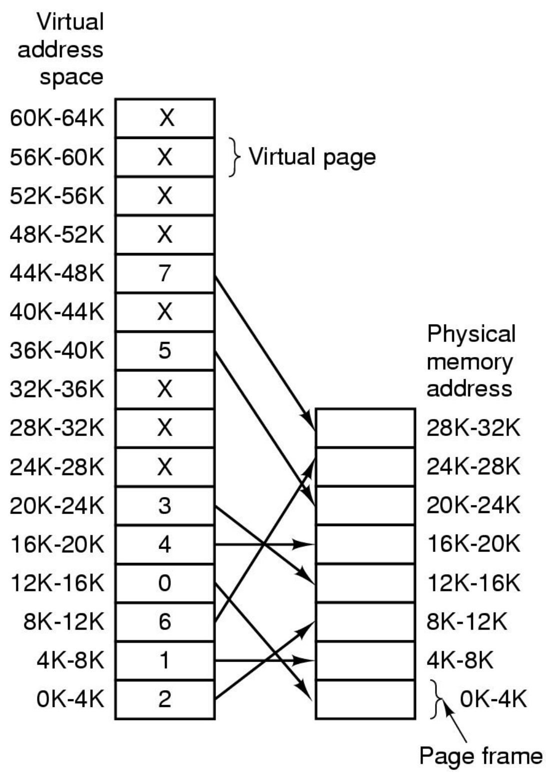
Problem 2:

For each of the following decimal virtual addresses, compute the virtual page number and offset for a 4 KB page and for and 8 KB page: 20000, 32768, 60000.

4 KB = 2^12 B
8 KB = 2^ 13 B

Address	Page Number (4KB)	Offset (4KB)	Page Number (8KB)	Offset (8KB)
20000	4	3616	2	3616
32768	8	0	4	0
60000	14	2656	7	2656

Problem 3:



Consider the page table of the figure. Give the physical address corresponding to each of the following virtual addresses:

(a) 29: Physical address: 8K + 29 = 8221
(b) 4100: Physical address: 4K + (4100 – 4K) = 4100
(c) 8300: Physical address: 24K + (8300 – 8K) = 24684

Problem 4:

A machine has 48 bit virtual addresses and 32 bit physical addresses. Pages are 8 KB. How many entries are needed for the page table?

Page size = 8 KB = 2^13 B
Offset = 13 bits
# of virtual pages = 2^(48 – 13) = 2^35 = # of entries in page table

Problem 5:

Consider a machine such as the DEC Alpha 21064 which has 64 bit registers and manipulates 64-bit addresses. If the page size is 8KB, how many bits of virtual page number are there? If the page table used for translation from virtual to physical addresses were 8 bytes per entry, how much memory is required for the page table and is this amount of memory feasible?

Page size = 8 KB = 2^13 B
Offset = 13 bits
Bits for virtual page number = (64 – 13) = 51
# of page table entries = 2^51
Size of page table = 2^51 * 8 B =2^54 B = 2^24 GB

Problem 6:

A computer with a 32-bit address uses a two-level page table. Virtual addresses are split into 9-bit top-level page table field, an 11 bit second-level page table field, and an offset. How large are the pages and how many are there in the address space?

Offset = 32 – 9 – 11 = 12 bits
Page size = 2^12 B = 4 KB
Total number of pages possible = 2^9 * 2^11 = 2^20

Problem 7:

Fill in the following table:

Virtual Address (bits)	Page Size	# of Page Frames	# of Virtual Pages	Offset Length (bits)	Addressable Physical Memory
16	256 B = 2^8	2^2	2^8	8	2^10 = 1 KB
32	1 MB = 2^20	2^4	2^12	20	2^24 = 16 MB
32	1 KB = 2^10	2^8	2^22	10	2^18 = 256 KB
64	16 KB = 2^14	2^20	2^50	14	2^34 = 16 GB
64	8 MB = 2^23	2^16	2^41	23	2^39 = 512 GB

Problem 8:

Fill in this table with the correct page evictions. Physical memory contains 4 pages.

Page Accesses	0	1	2	3	4	1	3	4	4	5	3	1	2	0	4	5	4
Optimal	-	-	-	-	0	-	-	-	-	4	-	-	-	3	2	-	-
FIFO	-	-	-	-	0	-	-	-	-	1	-	2	3	4	5	1	-
LRU	-	-	-	-	0	-	-	-	-	2	-	-	4	5	3	1	-
LFU	-	-	-	-	0	-	-	-	-	2	-	-	5	2	-	0	-
MRU	-	-	-	-	3	-	1	-	-	4	-	3	-	-	0	-	-