

4-14.

分配算法	请求	空闲区链表
首次适应	12K	10K, 28K , 20K, 18K, 7K, 9K, 12K, 15K
	10K	28K, 20K, 18K, 7K, 9K, 12K, 15K
	9K	19K , 20K, 18K, 7K, 9K, 12K, 15K
最佳适应	12K	10K, 40K, 20K, 18K, 7K, 9K, 15K
	10K	40K, 20K, 18K, 7K, 9K, 15K
	9K	40K, 20K, 18K, 7K, 15K
最坏适应	12K	10K, 28K , 20K, 18K, 7K, 9K, 12K, 15K
	10K	10K, 18K , 20K, 18K, 7K, 9K, 12K, 15K
	9K	10K, 18K, 11K , 18K, 7K, 9K, 12K, 15K

4-15. (1) $2 \times 4096 + 20 = 8212$

(2) $1 \times 4096 + 1004 = 5100$

(3) $6 \times 4096 + 108 = 24684$

(4) 页号越界

4-16.

算法	页访问序列													
	0	1	7	2	3	2	7	1	0	3	2	5	1	7
FIFO	0	0	0	0	3	3	3	3	3	3	3	3	3	7
		1	1	1	1	1	1	1	0	0	0	0	0	0
			7	7	7	7	7	7	7	7	7	5	5	5
				2	2	2	2	2	2	2	2	2	1	1
	缺页：9次													
LRU	0	0	0	0	3	3	3	3	0	0	0	0	1	1
		1	1	1	1	1	1	1	1	1	1	5	5	5
			7	7	7	7	7	7	7	7	2	2	2	2
				2	2	2	2	2	2	3	3	3	3	7
	缺页：11次													
OPT	0	0	0	0	3	3	3	3	3	3	3	3	3	3
		1	1	1	1	1	1	1	1	1	1	1	1	1
			7	7	7	7	7	7	0	0	0	0	0	7
				2	2	2	2	2	2	2	2	5	5	5
	缺页：8次													

4-17. (1) $219 + 430 = 649$

(2) $2300 + 10 = 2310$

(3) 段内越界

(4) $1326 + 400 = 1726$

(5) 段内越界

4-18.

(1) 物理页框为 4096B，65536B 的存储空间可划分为 16 个物理页框，该主存空间不适合此进程。

(2) 物理页框为 512B，65536B 的存储空间可划分为 128 个物理页框，在仅考虑该进程所需占用存储空间的情况下，该主存空间可以满足此进程的请求。

4-19. 9 位表示页号，10 位表示页内地址。

4-20.

(1) 程序 A 缺页 50 次，程序 B 缺页 $100 \times 50 = 5000$ 次。

(2) 程序 A 缺页 100 次，程序 B 缺页 $100 \times 100 = 10000$ 次。

4-21. 根据题目所述含义，在快表与页表并行访问时，需考虑：

(1) 页面不失效，即页面在内存中，此种情况又可细分如下：

a) 快表命中，直接形成物理地址；

b) 快表未命中，需访问页表，此时需区分：

(2) 页面失效，此时需先访问快表和页表，才能发现页面失效，产生缺页中断，进行缺页处理时需在内外存之间传送页面，并更新页表和快表，更新页表为一次内存写操作，更新快表较快，可与更新页表同时完成；

(3) 地址变换完成后，访问物理内存。

基于上述分析，内存有效存取时间为：

$$(1-0.1) \times [0.75 \times 1 + (1-0.75) \times 8] + 0.1 \times (8 + 5000 + 8) + 8 = 512.075$$

4-23. 一个进程理论上可具有 $2^{64}B$ 的逻辑地址空间，一个物理块可存放 $8192/4 = 2^{11}$ 个页表项，因此若要求页表信息存放在一个物理块中，应采用 5 级页表(7+11+11+11+11+13)。