

10.1 现有一个内存空间分配器，采用伙伴算法。假设物理内存总共 64 KB，

- 1. 请给出第一级的一对伙伴块的起始地址
- 2. 请给出第二级的二对伙伴块的起始地址
- 3. 地址 0xa700，已知它位于第 7 级伙伴块中，请问该块的伙伴块的起始地址

- 1. 默认伙伴块起始物理地址位0x0000。
第一级伙伴块地址分别为：0x0000和0x8000
- 2. 第二级伙伴块一共有两对。第一对起始地址为：0x0000和0x4000。第二对起始地址为：0x8000和0xc000。
- 3. 地址：0xa700

下表左闭右开：

	起始地址	终止地址	大小
第一级	0x8000	0x10000	32K
第二级	0x8000	0xc000	16K
第三级	0xa000	0xc000	8K
第四级	0xa000	0xb000	4K
第五级	0xa000	0xa800	2K
第六级	0xa400	0xa800	1K
第七级	0xa600	0xa800	512B

伙伴块为[0xa400,0xa600)，起始地址0xa400.

10.2 假设一个程序对其地址空间中虚页的访问序列为0,1,2, ...,511,422,0,1,2,...,511, 333,0,1,2,...，即访问一串连续地址（页0到页511）后会随机访问一个页（页422或页333），且这个访问模式会一直重复。请分析说明：

- 1. 假设操作系统分配给该程序的物理页框为500个，那么，LRU，Second Chance和FIFO这三种算法中哪一个会表现较好（即提供较高的缓存命中率），或是这三种算法都表现不佳？为什么？

	LRU	Second Chance	FIFO
命中率	1/513~3/513	1/513~3/513	1/513~3/513

三种算法表现都不佳。有趣的是，一种非常好的算法是不进行替换。可以确定不进行替换的缓存命中率是 $\frac{501}{513}$ ，这比三种算法都要好。这是因为这个访问模式并不满足大多数进程所拥有的局部性访问特性，其循环遍历所有的页框，这使得传统的置换算法都不能很好的适应这个场景。