

1. (1) NRU 演算法優先置換未被使用、未被修改或是被修改時間最久的 page，因此置換 page 0（page 0 的 R 和 M 位元都是 0，代表未被使用、修改）。
- (2) FIFO 演算法優先置換最早被載入的 page，因此置換 page 2。
- (3) LRU 演算法優先置換最久未被使用的 page，因此置換 page 1。
- (4) second chance 演算法和 FIFO 演算法相似，若最早被載入的 page 有被使用過，則優先置換未被使用的 page，因此置換 page 0（第二早被載入且未被使用）。

2.

Page	Loaded	R	M
A	18	1	1
C	23	0	1
G	5	1	1
H	7	1	0
B	32	0	0
L	19	1	0
N	3	1	1
D	8	0	1

second chance 演算法和 FIFO 演算法相似，若最早被載入的 page 有被使用過，則優先置換未被使用的 page，因此置換 page D（第四早被載入且未被使用）。

3. 實體記憶體使用的是實體位址（physical address），而 process 使用的是虛擬位址（virtual address）。

虛擬位址是連續的（讓 process 以為有連續的可用記憶體，但對應到的實體記憶體位址可能是分散的），虛擬記憶體也允許 process 不受實體記憶體限制而使用更多記憶體。

4. 兩個方法選擇的 page 一樣，只是表示方法不同。

clock	形成循環的 FIFO，只有 tail 指標 表示 head 的方法：tail->next
second chance	FIFO，有 head 和 tail 兩個指標 表示 head 的方法：head

5.

page 0	01101110
page 1	01001001
page 2	00110111
page 3	10001011