1. 在一系列栈操作序列中加入MULTIPUSH操作，MULTIPUSH一次性将*k*个元素压入栈中.试问加入MULTIPUSH后栈操作的平摊代价是否仍为*O*(1)?

~~原答案：是~~*~~O~~*~~（1）。设栈长度为~~*~~n，~~*~~进行了~~*~~n~~*~~次操作。我们设push进行了a次，multipush进行了b次，pop进行了c次。那么有~~

~~设multipush每一次送进栈的个数分别为。那么我们有均摊代价T为~~

~~我们分析的上界。由于对于而言，有，这是因为在第~~*~~i~~*~~次进行multipush的时候栈已经被占用了个，至多释放pop的次数个，即~~*~~c~~*~~个。所以我们有 所以对于b我们有~~

~~即~~

~~所以根据(2)我们有~~

~~证毕。~~

后来听说栈长度无限，更正如下:

我们考虑极端情况的开销。考虑操作序列中每次操作都是巨大开销的MULTIPUSH和MULTIPOP，那么我们有：每次MULTIPUSH 和 MULTIPOP的开销都是O(n)级别的，因此总操作代价是O(n2),平摊代价为O(n)，所以不是O(1)了。

2. 举例说明如果在k位二进制计数器中加入DECREAMENT操作(减1)，则*n*个操作的时间开销能够达到(*nk*).

假设A=(0000000…0)2为k位二进制数，定义操作序列为op=[-1,1,-1,1,-1,1,…,(-1)n],表示n个操作，-1表示减1操作，1表示加1操作，那么对于这个例子的A,每次操作都需要k个单位时间代价。因此时间开销达到了