We are given the following layer partition, for which we request page 3:

a) layer partition

layers	L_0	L_1	L_2	L_3	L_4	L_5
pages	1,2	3,4	5,6	7,8	9,10	11,12
created at time t	—	30	40	50	60	70

b) request(3) yields:

layers	L_0	L_1	L_2	L_3	L_4	L_5
pages	1,2,4	5,6	7,8	9,10	∅	11,12,3
created at time t	—	40	50	60	—	70

The contents of layers $L_1 \setminus \{3\}$, L_2, L_3 and L_4 all have to move one position to the left.

Let us see how this is implemented by TStamp:

TStamp

a) birth array

	L_1	L_2	L_3	L_4	L_5
array index	0	1	2	3	4
t	30	40	50	60	70
v	1	1	1	1	1
prefix sum of v	1	2	3	4	5

b) request(3)

array index	0	1	2	3	4
t	X	40	50	60	70
v	X	1	1	1	1
prefix sum of v	X	2	3	4	5

Column 0 is deleted (indicated by X).

Columns 1 to 4 are moved one position to the left (this becomes obvious if we look at the t values).

request(3) yields:

array index	0	1	2	3	4
t	40	50	60	70	—
v	1	1	1	2	—
prefix sum of v	1	2	3	5	—

v is updated to indicate the insertion of an empty set, and the prefix sum is recomputed.

L_1 L_2 L_3 L_5

We can convince ourselves that this reflects the structure of the layer partition.

Recall that the birth array has a maximum length of k. In such a case k-1 columns have to be shifted to the left which is expensive.