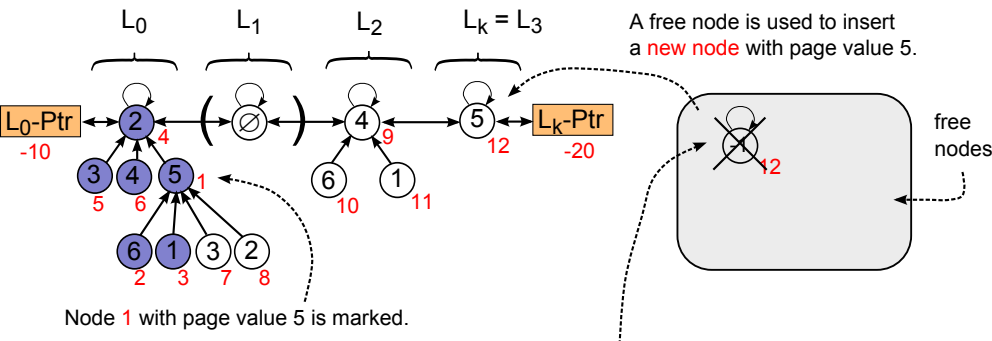
page 5 is requested:



address array

page value	1	2	3	4	5	6
node id	11	8	7	9	12	10

page value 5 is now in the newly inserted node 12:
address[5]=12

The queue tells us that node 12 was free:
Q = [~~12~~]

A dequeue yields an empty queue:
Q = [ ]

node id	1	2	3	4	5	6	7	8	9	10	11	12
page value	5	6	1	2	3	4	3	2	4	6	1	5
parent	4	1	1	4	4	4	1	1	9	9	9	12
left	-1	-1	-1	-10	-1	-1	-1	-1	4	-1	-1	9
right	-1	-1	-1	9	-1	-1	-1	-1	12	-1	-1	-20
count	0	0	0	0	0	0	0	0	1	0	0	0
marked	1	1	1	1	1	1	0	0	0	0	0	0
rank	1	0	0	2	0	0	0	0	1	0	0	0

The variable Lk-Ptr is updated:

L0-Ptr = 4
Lk-Ptr = 12

We mark node 1 by setting marked[1]=1.

Column 12 describes the newly inserted node 12.

There are no free nodes which becomes apparent if we look at the empty queue. 6 nodes are unmarked and the other 6 nodes are marked. The problem is whenever we request pages from L_0 and L_i with $0<i<k$, we need a free node from the queue. In order to have free nodes again we will have to call the cleanup procedure which removes the 6 marked nodes and puts them into the queue again.