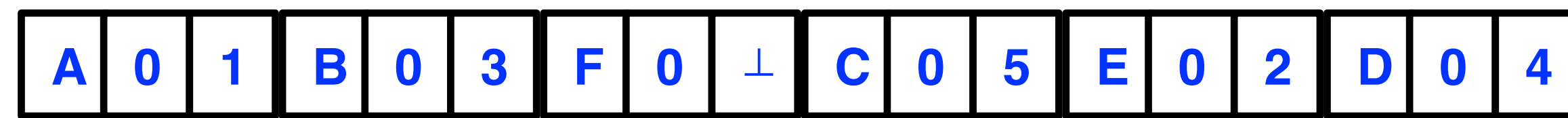


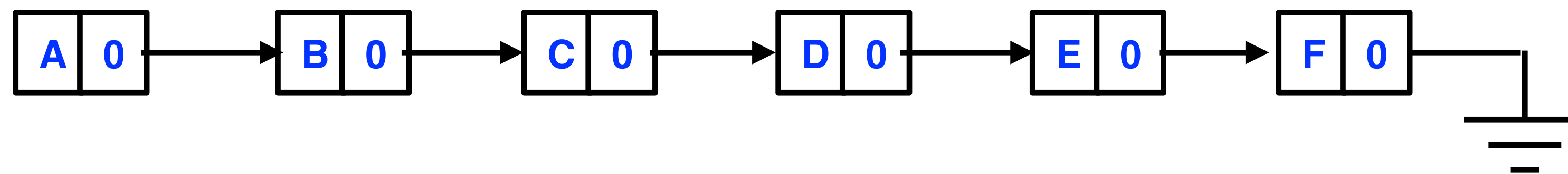
# Worksheet: Pointer Skipping

You are given a linked list, and you need to compute the *rank* of each element of the list, i.e. the distance of that element from the end of the list.

Give a high-level idea of how would you solve this problem in parallel using pointer skipping. You can assume that the list is stored in a contiguous array, with a pointer to the next element in the list being a simple index of that element. For example, the following array:



Represents the following list:

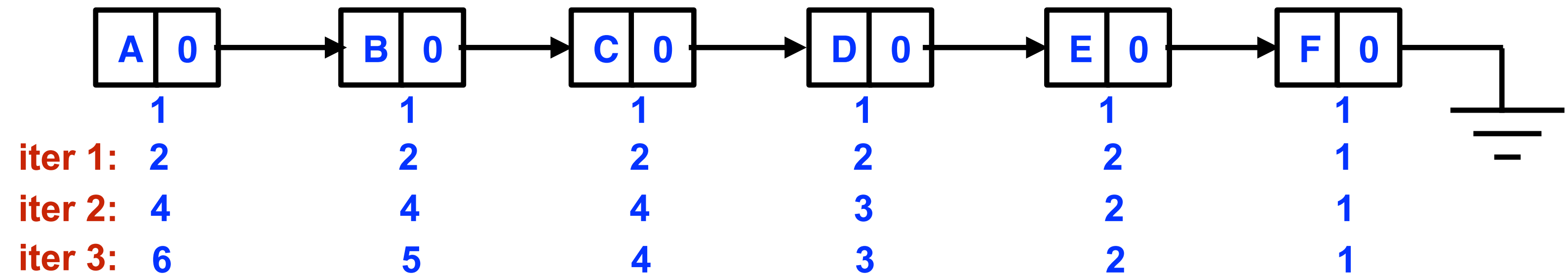


What is the total WORK that your solution would perform (integer addition counts as WORK(1), everything else is ignored)?



# Worksheet: Pointer Skipping

Assume  $d[i] = 1$  for all nodes,  
current val before update for succ[i]



Algorithm:

1. Repeat  $\log N$  times:

1. Finish

1. Create an async task for each list node

2. In each task i:

1. set  $d[i] += d[\text{succ}[i]]$

2. set  $\text{succ}[i] = \text{succ}[\text{succ}[i]]$

What is the big-O for total WORK that your solution would perform (integer addition counts as WORK(1), everything else is ignored)?

**WORK =  $O(N \log N)$ , CPL =  $O(\log N)$**

