Partition Linked List

input: 5

original list:

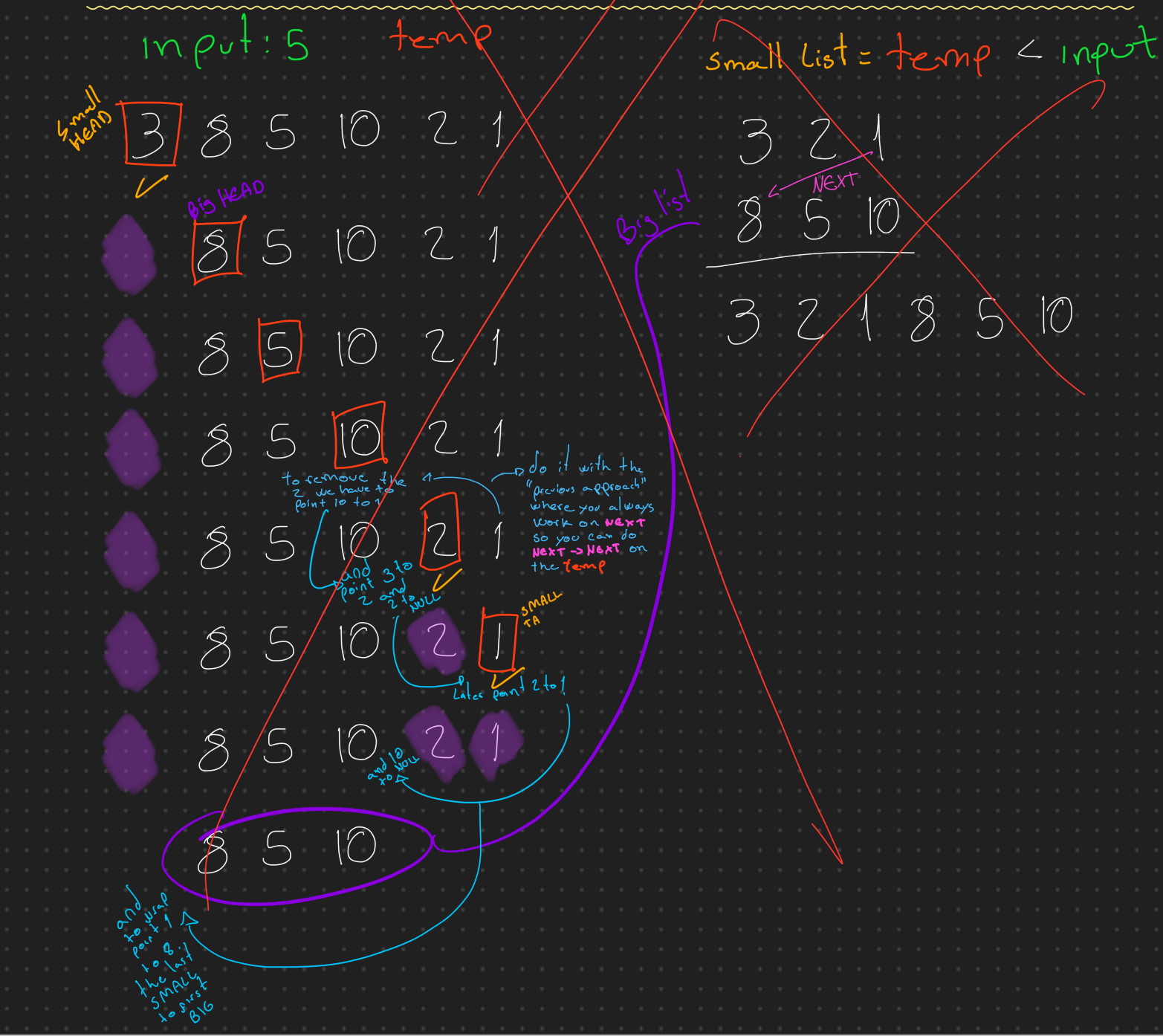
3 8 5 10 2 1

expected output

3 2 1 8 5 2

everything that is lower than the input goes to the start of the list.

we can create 2 LL and then join it.



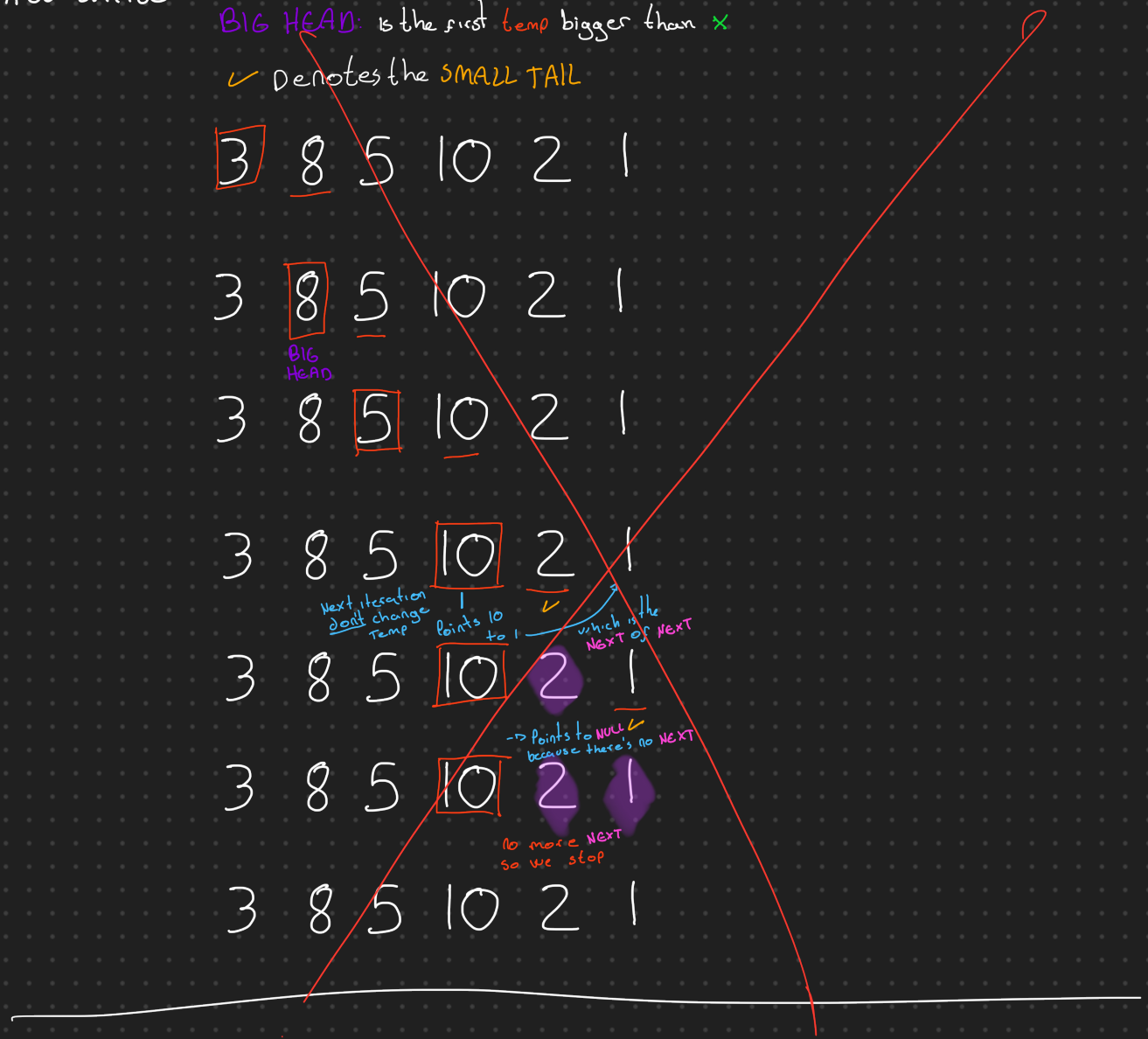
GOAL:

REMOVE ALL SMALL

X = 5 SMALL: <X

BIG HEAD: is the first temp bigger than X

↙ Denotes the SMALL TAIL



No Luck

I gave up and watched an explanation.

my code and approach worked for like 90% of the cases but not all.

create two dummy HEAD
↳ prev

X = 5

current
↳ just iterate normally
do not modify it.
Point current to SMALL or BIG prev NEXT

SMALL prev → 0 → 3 2 1 → set SMALL prev NEXT to BIG DUMMY NEXT to merge these new lists

BIG prev → 0 → 8 5 10 ⚠

{ at this moment 10 NEXT points to 2. Set BIG prev to NULL to define it as our new TAIL

↳ the function returns SMALL DUMMY NEXT which skips the 0