1D size k

① Stack: O(n) 可解.

② Dedup O(n·(n/k))

while (length ≠ size)

length = s.size()

for(i in length):

if i==0 || s[i]≠s[i-1]

count=1

else

s = s[i-k : i]

break

③ count:

if count[i] = k

s = [:i-k+1] +

s[i+1:]

1D size k+ (zuma)

Stack O(n) 可解.

s = s+"$"

① if not stack

append

② 相同 update

③ 不同

① 没到 k 以上? append

② 到 k 了

pop:

① 空了不样 append

② 一样 update.

2D 如何不用 mark negative?

1) 存入 set 中, 将需要 remove 掉的 设为 0, 然后做 Gravity Pull

Runtime? Scan O(N²·(N²/k)) 每次 crush k个.

Recursion 1Loop: (这是 k+ 版本)

Dedup O(n·(n/k))

Dedup: s = s+"$"

prevstart = 0

for i,c in enumerate(board):

if c != board[prevstart]:

if i - prevstart >= 3:

return dedup(board[:prevstart] + board[i:])

prevstart = i

return board