Problem # 870: Advantage Shuffle (Medium)

----------------------------------------------

<https://leetcode.com/problems/advantage-shuffle/>

My Solution:

<https://leetcode.com/problems/advantage-shuffle/discuss/1126603/My-Python-Solution-only-passed-58-out-of-67-test-cases-Please-explain>

My Solution:

1. Sort array A and store it in sorted\_A.
2. Sort array B and store it in sorted\_B.
3. Initialize an empty dictionary to hold mapping values.
4. Let n be the length of A
5. Initialize i to n-1, j to 0 and k to n-1, where i points to the end of sorted\_A , j points to the beginning of sorted\_A and k points to the end of sorted\_B.
6. Now traverse sorted\_B from the end to the beginning. If the index at sorted\_B is more than the index at sorted\_A, we have lost this already. So give the dictionary mapping the smallest value in sorted\_A that is available which will be at the beginning and advance the pointer j by 1. Otherwise, map sorted\_B element to the largest value of A that is available from the end and decrement pointer B by 1. We always need to decrement i by 1 as we are traversing sorted\_B array from end to beginning.
7. Initialize result with an empty array to hold the shuffled A array.  
   Traverse B array from beginning to end and for each element get the mapping from my\_dict. Append the value from the dictionary to the result list.

My solution failed for test case 58.

Input:[2,0,4,1,2]  
[1,3,0,0,2]

Output:[2,4,1,1,0]

Expected:[2,0,2,1,4]

You will see that of the 5 elements in A, A[i] > B[i] for 4 elements in my result (i.e. output) and also in the expected answer. So I think my answer should be accepted.

Please can someone explain to me what is wrong with my solution.

sorted\_A = sorted(A)

sorted\_B = sorted(B)

my\_dict = {}

n = len(A)

i = n-1

j = 0

k = n-1

while (i >= 0):

if sorted\_B[i] >= sorted\_A[k]:

my\_dict[sorted\_B[i]] = sorted\_A[j]

j += 1

else:

my\_dict[sorted\_B[i]] = sorted\_A[k]

k -= 1

i -= 1

result = []

for i in range(n):

result.append(my\_dict[B[i]])

return result