

# Q1

For this question, we start with input and value all the variables respectively, then using for loop, we make it possible to continuously input and output. We then handle the question using three functions: update array, calculate sum and find distinct. For the third one, it is obvious that we have to consider the number of the element with the same absolute value (if its num  $\geq 2$ , there are 2 distinct values except for 0).

The main problem when we program is the time efficiency, so I make 2 functions into the main program, we first calculate the sum and after that, we only add the difference between the new element and the old ones. The update array function is also removed. For the distinct value function, I change the algorithm. At first I use sort function and scan the new list to find distinct value ( $O(n \log n)$ ), then I come up with the idea of dictionary, by using it, we make absolute value of element keys and its number values, then we not only make it more efficient ( $O(n)$ ), but also when we update the array, we simply update the dictionary, so we do not need to scan the array again and again.

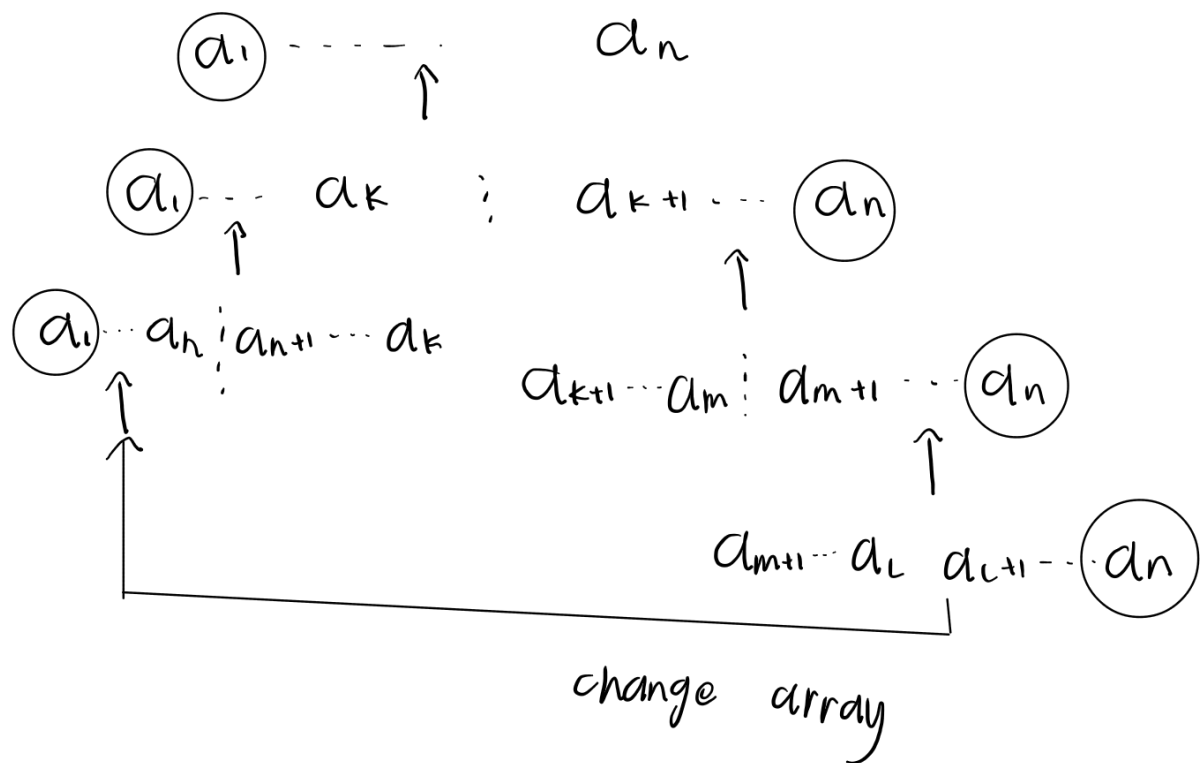
In short, I think my program has an advantage in its use of dictionary. By using it, we promote time efficiency not only in the first progress of finding distinct values in original arrays, but also

make it easier to update the distinct value.

## Q2

For the question, we also handle the input first. After all list assigned, we find the answer and the key points, since the question do not involve that much with time efficiency.

I do it by drawing a mindmap.



As you can see, the first element of left and right list always equals that of the permutation given. Also, the original location of the element in left list in the permutation must smaller then that of the right list (I actually almost ignore the obvious condition). Third, if we split the same list, there must be same element in either left or right list, then we check their location and the

previous ones and make our choice. Last, when we change the list we split our previous consumption fail, but the same number must have been put out in the previous location, so we can make our choice.

In short, I think my method has an advantage in thinking, by plotting the mind map and thinking from top to bottom, we can solve the problem easily.