AFP - Assignment 2 - Stack Permutation

Maximo Garcia Martinez

1. Algorithm

1. Reverse both Input (I) and Output (O) lists. Create a new empty list called Stack (S).
2. If I, O and S are empty return true.
3. If the head of S and the top of O are equals:
   - If S and I are empty and O is not, return false.
   - Otherwise, execute the algorithm again from step 2, with I, the tail of S and the tail of O.
4. If the head of I and the top of O are equals:
   - If S and I are empty and O is not, return false.
   - Otherwise, execute the algorithm again from step 2, with S, the tail of I and the tail of O.
5. Append the head of I into S. Go to step 2.

2. Testing

I have list with different examples. The type of this list is `[([a1], [a2], Bool)]`. The first element of the tuple works as Input, the second as Output and the third is a Bool that the algorithm should return. An example of this list can be seen here:

```
[
    ([1,2],[2,1],True),
    ([],[1],False),
    ([3,2,1],[1,3,2],True),
    ([1..1000000],[1..1000000], True)
]
```

I am testing with input and output of the same size, to see if the basic implementation works. Then, I added some test if having Inputs and Output of different sizes provokes an error or not and finally, I added some tests with lists with one million elements and test the efficiency of my program.