

341. Flatten Nested List Iterator

You are given a nested list of integers `nestedList`. Each element is either an integer or a list whose elements may also be integers or other lists. Implement an iterator to flatten it.

Implement the `NestedIterator` class:

- `NestedIterator(List<NestedInteger> nestedList)` Initializes the iterator with the nested list `nestedList`.
- `int next()` Returns the next integer in the nested list.
- `boolean hasNext()` Returns true if there are still some integers in the nested list and false otherwise.

Your code will be tested with the following pseudocode:

```
initialize iterator with nestedList
```

```
res = []
```

```
while iterator.hasNext()
```

```
    append iterator.next() to the end of res
```

```
return res
```

If `res` matches the expected flattened list, then your code will be judged as correct.

Example 1:

Input: `nestedList = [[1,1],2,[1,1]]`

Output: `[1,1,2,1,1]`

Explanation: By calling `next` repeatedly until `hasNext` returns false, the order of elements returned by `next` should be: `[1,1,2,1,1]`.

Example 2:

Input: `nestedList = [1,[4,[6]]]`

Output: `[1,4,6]`

Explanation: By calling next repeatedly until hasNext returns false, the order of elements returned by next should be: [1,4,6].

Constraints:

- $1 \leq \text{nestedList.length} \leq 500$
- The values of the integers in the nested list is in the range $[-10^6, 10^6]$.

```
class NestedIterator(object):

    def __init__(self, nestedList):
        self.stack = [[nestedList, 0]]

    def next(self):
        self.hasNext()
        nestedList, i = self.stack[-1]
        self.stack[-1][1] += 1
        return nestedList[i].getInteger()

    def hasNext(self):
        s = self.stack
        while s:
            nestedList, i = s[-1]
            if i == len(nestedList):
                s.pop()
            else:
                x = nestedList[i]
                if x.isInteger():
                    return True
                s[-1][1] += 1
                s.append([x.getList(), 0])
        return False
```