Daily Coding Problem #164

Problem

This problem was asked by Google.

You are given an array of length `n + 1` whose elements belong to the set `{1, 2, ..., n}`. By the pigeonhole principle, there must be a duplicate. Find it in linear time and space.

Solution

One method to solve this is to iterate over the array and look in location `i` of the array: if `lst[i]` holds `i`, then keep going. If `lst[i]` holds `j`, then swap `lst[i]` and `lst[j]` and repeat until it's the correct value. If we encounter the same value at `lst[j]` then we have found our duplicate.

```python
def duplicate(lst):
    i = 0
    while i < len(lst):
        if lst[i] != i:

            j = lst[i]
            if lst[j] == lst[i]:
                return j
            lst[i], lst[j] = lst[j], lst[i]
        else:
            i += 1
    raise IndexError('Malformed input.')
```

This runs in O(n) time and constant space.

We can also simply sum up all the elements in the array and subtract it by the sum of 1 to n, using the formulas n * (n + 1) / 2. We should be left with the duplicate.

```python
def duplicate(lst):
    n = len(lst) - 1
    return sum(lst) - (n * (n + 1) // 2)
```

This takes O(n) time and O(1) space.