4.18. Palindrome-Checker

An interesting problem that can be easily solved using the deque data structure is the classic palindrome problem. A **palindrome** is a string that reads the same forward and backward, for example, *radar*, *toot*, and *madam*. We would like to construct an algorithm to input a string of characters and check whether it is a palindrome.

The solution to this problem will use a deque to store the characters of the string. We will process the string from left to right and add each character to the rear of the deque. At this point, the deque will be acting very much like an ordinary queue. However, we can now make use of the dual functionality of the deque. The front of the deque will hold the first character of the string and the rear of the deque will hold the last character (see Figure 2).

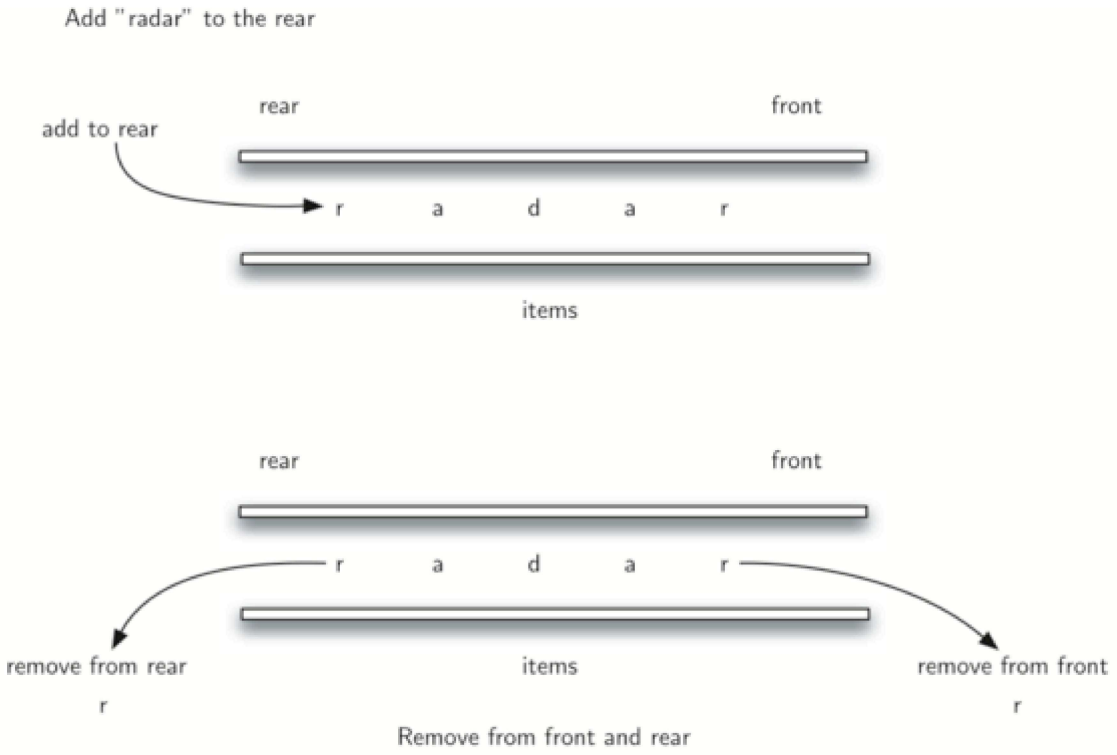

Figure 2: A Deque

Since we can remove both of them directly, we can compare them and continue only if they match. If we can keep matching first and the last items, we will eventually either run out of characters or be left with a deque of size 1 depending on whether the length of the original string was even or odd. In either case, the string must be a palindrome. The complete function for palindrome-checking appears in ActiveCode 1.

Run Load History

```
1 from pythonds.basic import Deque
2
3 def palchecker(aString):
4     chardeque = Deque()
5
6     for ch in aString:
7         chardeque.addRear(ch)
8
```

(ImplementingaDequeinPython.html) (Lists.html)

```
 9     stillEqual = True
10
11     while chardeque.size() > 1 and stillEqual:
12         first = chardeque.removeFront()
13         last = chardeque.removeRear()
14         if first != last:
```

Activity: 4.18.1 A Palindrome Checker Using Deque (palchecker)

You have attempted 1 of 2 activities on this page

user not logged in

(ImplementingaDequeinPython.html)

(Lists.html)