数据结构与算法（Python版）

双端队列抽象数据类型及Python实现

陈斌 北京大学 gischen@pku.edu.cn

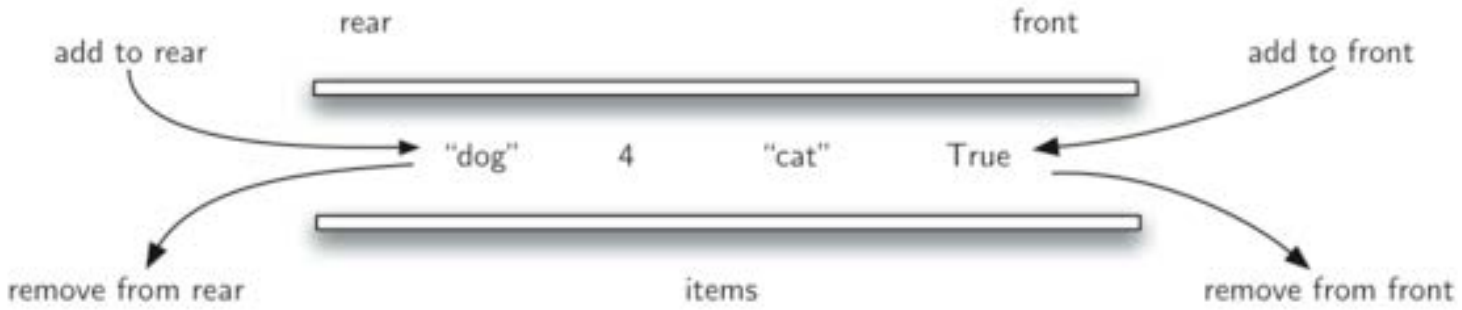
双端队列Deque：什么是Deque?

- **双端队列Deque是一种有次序的数据集，**

  跟队列相似，其两端可以称作“首”“尾”端，

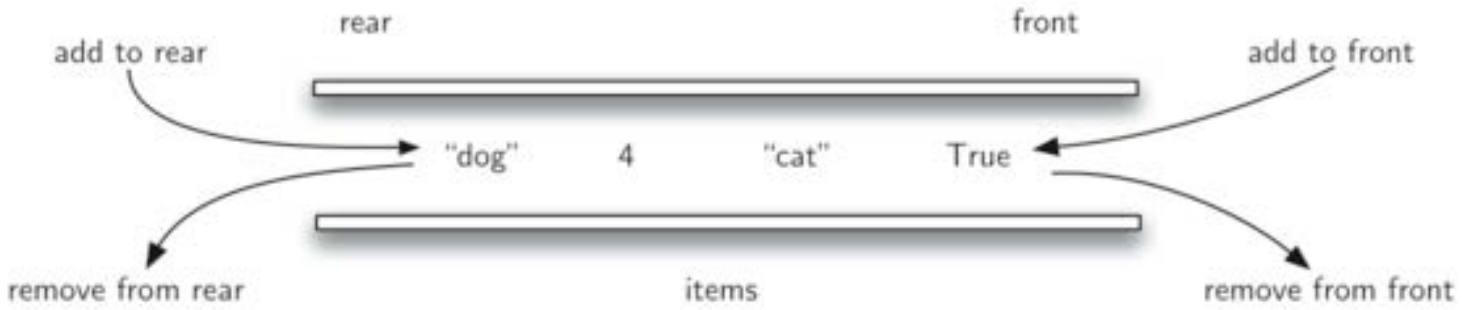
  但deque中数据项既可以从队首加入，也可以从队尾加入；数据项也可以从两端移除。

  某种意义上说，双端队列集成了栈和队列的能力



双端队列Deque：什么是Deque?

- 但双端队列并不具有内在的LIFO或者FIFO特性
  - 如果用双端队列来模拟栈或队列
  - 需要由使用者自行维护操作的一致性



抽象数据类型Deque

- deque定义的操作如下：

  Deque()：创建一个空双端队列

  addFront(item)：将item加入队首

  addRear(item)：将item加入队尾

  removeFront()：从队首移除数据项，返回值为移除的数据项

  removeRear()：从队尾移除数据项，返回值为移除的数据项

  isEmpty()：返回deque是否为空

  size()：返回deque中包含数据项的个数

抽象数据类型Deque

双端队列操作	双端队列内容	返回值
d=Deque()	[]	Deque object
d.isEmpty()	[]	True
d.addRear(4)	[4]	
d.addRear('dog')	['dog',4,]	
d.addFront('cat')	['dog',4,'cat']	
d.addFront(True)	['dog',4,'cat',True]	
d.size()	['dog',4,'cat',True]	4
d.isEmpty()	['dog',4,'cat',True]	False
d.addRear(8.4)	[8.4,'dog',4,'cat',True]	
d.removeRear()	['dog',4,'cat',True]	8.4
d.removeFront()	['dog',4,'cat']	True

Python实现ADT Deque

- 采用List实现
  - List下标0作为deque的尾端
  - List下标-1作为deque的首端
- 操作复杂度
  - addFront/removeFront O(1)
  - addRear/removeRear O(n)

```
class Deque:
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return self.items == []

    def addFront(self, item):
        self.items.append(item)

    def addRear(self, item):
        self.items.insert(0,item)

    def removeFront(self):
        return self.items.pop()

    def removeRear(self):
        return self.items.pop(0)

    def size(self):
        return len(self.items)
```

“回文词”判定

- “回文词”指正读和反读都一样的词

  如radar、madam、toot

  中文“上海自来水来自海上”

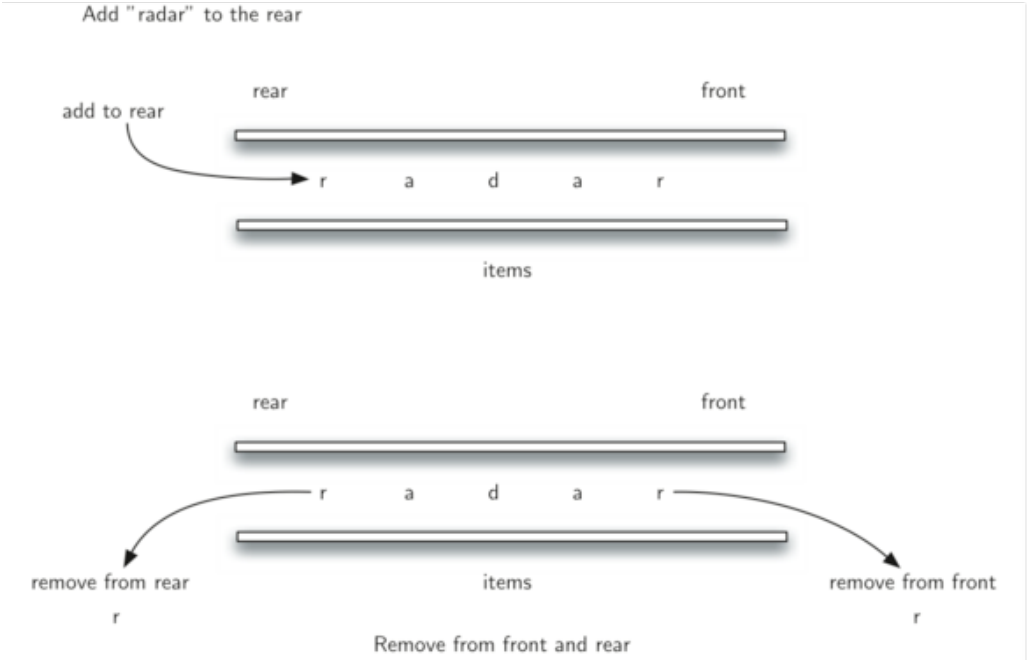
  “山东落花生花落东山”

“回文词”判定

- 用双端队列很容易解决“回文词”问题

  先将需要判定的词从队尾加入deque

  再从两端同时移除字符判定是否相同，直到deque中剩下0个或1个字符



“回文词”判定：代码


```
from pythonds.basic.deque import Deque

def palchecker(aString):
    chardeque = Deque()

    for ch in aString:
        chardeque.addRear(ch)

    stillEqual = True

    while chardeque.size() > 1 and stillEqual:
        first = chardeque.removeFront()
        last = chardeque.removeRear()
        if first != last:
            stillEqual = False

    return stillEqual

print(palchecker("lsdkjfskf"))
print(palchecker("radar"))
```