دانشکده علوم ریاضی و آمار

تهیه و تنظیم: محمدرضا ظاهری وعلیرضا کاظمی

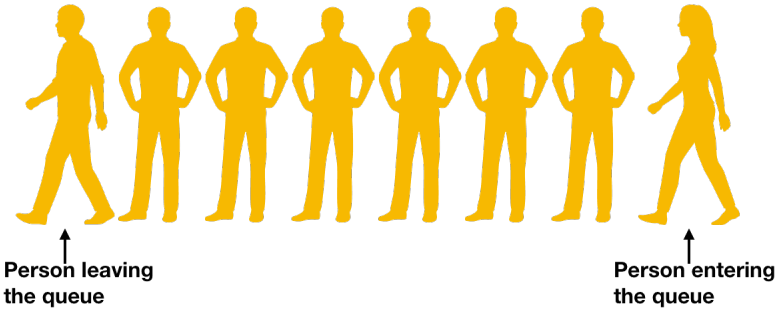
داده ساختار صف (Queue)

پاییز ۱۴۰۱

ساختمان داده‌ها و الگوریتم‌ها

۱ صف چیست؟

صف یک ساختار داده است که تا حدودی شبیه پشته محسوب می‌شود؛ اما بر خلاف پشته، صف از هر دو سمت باز است. از یک سمت همواره برای درج داده‌ها و از سمت دیگر برای حذف داده‌ها استفاده می‌شود. صف از روش FIFO)ورودی اول–خروجی اول) استفاده می‌کند . در این روش هر داده‌ای که اول در صف ذخیره شود، هنگام خروج نیز قبل از همه خارج می‌شود .

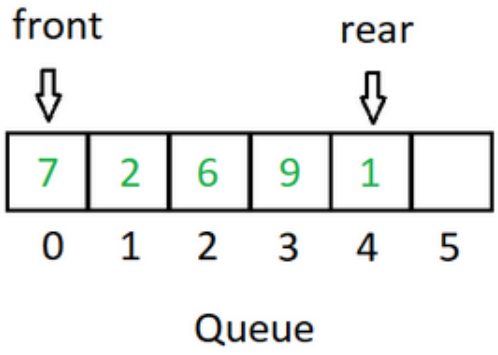


۲ کاربرد صف چیست؟

صف اولویت
صف نانوایی
سیستم های عامل
هر نوع صف دیگر در زندگی واقعی

۳ انواع صف چیست؟

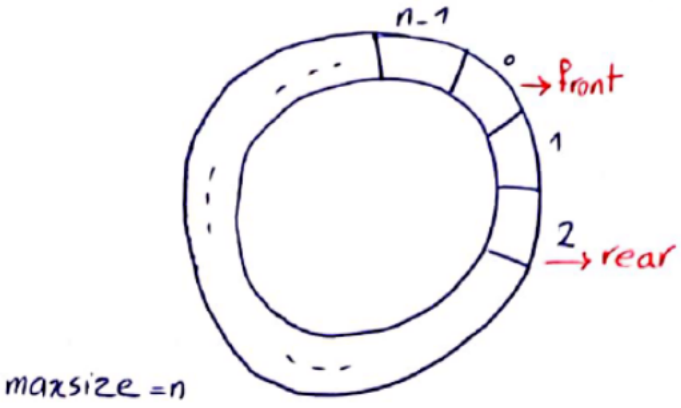
۱.۳ صف خطی



توجه

مشکل‌صف ساده: پس از انجام تعدادی عمل حذف و درج در صف، با اینکه آرایه فضای آزاد دارد، امکان درج عنصر جدید را نخوا هیم‌داشت.

۲.۳ صف حلقوی



۴ چگونه صف بسازیم؟

ابتدا به وسیله شی گرایی در بایتون یک کلاس صف ساختیم و برای آن عناصر rear ، front ، maxsize قرار دادیم .

```
class Queue:
    def __init__(self, max_size: int):
        self.max = max_size
        self.queue = [None for _ in range(max_size)]
        self.rear = self.front = 0

    @property
    def size(self):
        if self.is_full():
            return self.max
        return (self.max - self.front + self.rear) % self.max
```

۵ توابع و دستورات ساخته شده و استفاده شده در صف چیست؟

۱.۵ isfull

این عمل مشخص می کند که صف پر است یا خیر؟

```
  def is_full(self):
      return self.rear == self.front and
      self.queue[self.rear] is not None
```

هزینه آن از مرتبه $O(1)$ است

۲.۵ isnull

این عمل مشخص می کند که صف خالی است یا خیر؟

```
def is_null(self):
    return self.rear == self.front and
    self.queue[self.rear] is None
```

هزینه آن از مرتبه $O(1)$ است

۳.۵ Enqueue

این عمل عنصر جدید را به انتهای صف اضافه می کند .

```
Queue_Enqueue(data):

    if Queue is full:
        return "Queue is Full"

    Queue[rear] = data
    rear = rear + 1

End Queue_Enqueue
```

هزینه آن از مرتبه $O(1)$ است

۴.۵ Dequeue

این عمل اولین عنصر را از ابتدای صف حذف می کند .

```
Queue_Dequeue(data):

    if Queue is null:
        return "Queue is Null"

    Queue[front] = Null
    front = front - 1

End Queue_Dequeue
```

هزینه آن از مرتبه $O(1)$ است

reverse ۵.۵

این عمل صف را برعکس می کند به این معنا که عنصر ها جابجا می شوند . ابتدا عناصر درون یک بشته ریخته می شوند و بعد از آن درون صف ریخته می شود تا صف برعکس شود .

```
def reverse(self):
    stack = LifoQueue(self.max)
    size = self.size
    for i in range(size):
        stack.put(self.dequeue())
    for i in range(size):
        self.enqueue(stack.get())
```

هزینه آن از مرتبه $O(n)$ است

peek ۶.۵

این عمل عنصر ابتدایی صف را برای ما باز می گرداند .

```
def peek(self):
    return self.queue[self.front]
```

هزینه آن از مرتبه $O(1)$ است

clear ۷.۵

این عمل صف را از عنصر خالی می کند .طریقه ساخت آن به این صورت است که عناصر صف را آنقدر حذف می کند که صف خالی از عنصر شود .

```
def clear(self):
    for i in range(self.size):
        self.dequeue()
```

هزینه آن از مرتبه $O(n)$ است

join ۶

این عمل دو صف را به یکدیگر اضافه می کند .

```
def join(self,queue2):
    if queue2.size + self.size > self.max:
        print("Overflow!")
    else:
        for _ in range(queue2.size):
            self.enqueue(queue2.dequeue())
```