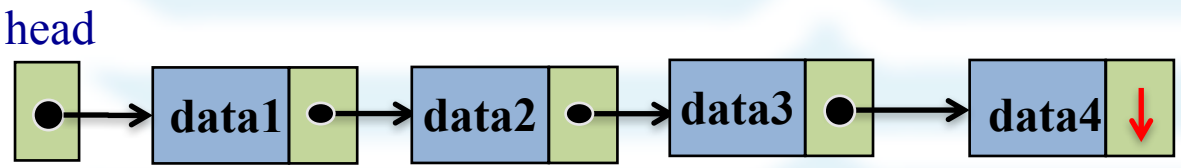
# پیاده‌سازی لیست تک پیوندی

- تابع درج یک گره در انتهای لیست (با داشتن داده مربوط به گره):

```
def Add2EndByValue(self, data):
```



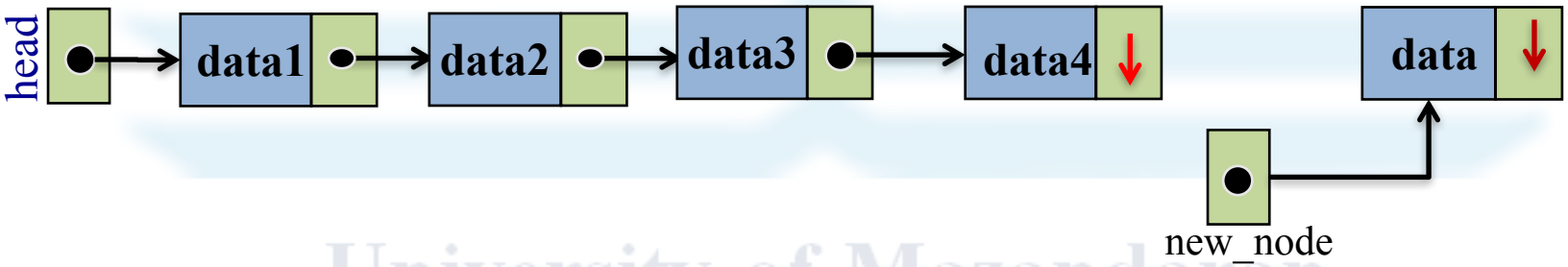

University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- **تابع درج یک گره در انتهای لیست (با داشتن داده مربوط به گره):**

```
def Add2EndByValue(self, data):
    new_node= self.SinglyLinkedListNode(data, None)
```

**گام۱:** تعریف گره




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- **تابع درج یک گره در انتهای لیست (با داشتن داده مربوط به گره):**

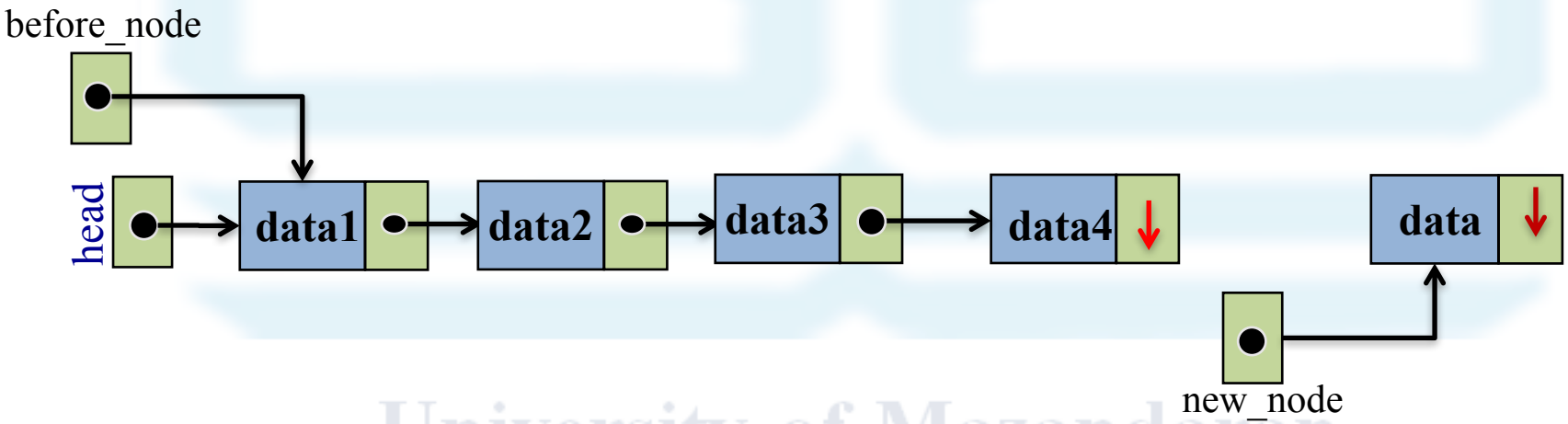
```
def Add2EndByValue(self, data):
    new_node= self.SinglyLinkedListNode(data, None)
    if self._size == 0:
        self.head = new_node
    else:
        before_node = self.head
```

**گام۱:** تعریف گره

**گام۲:** اضافه کردن گره به لیست

- دستورات لازم در صورتی که لیست خالی باشد:

- دستورات لازم در صورتی که لیست خالی نباشد:




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- **تابع درج یک گره در انتهای لیست (با داشتن داده مربوط به گره):**

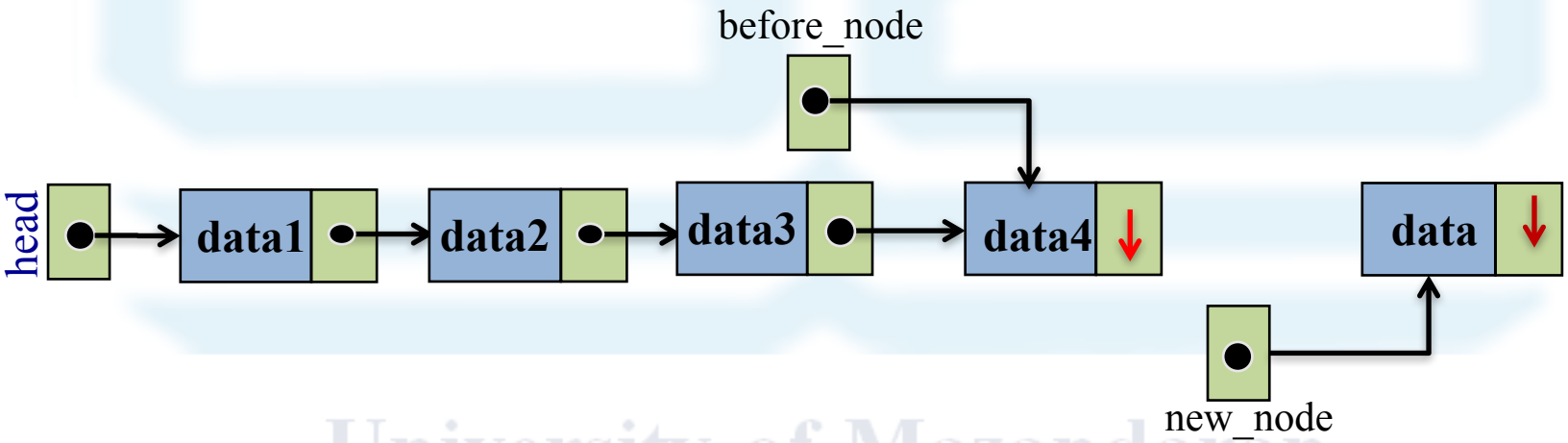
```
def Add2EndByValue(self, data):
    new_node= self.SinglyLinkedListNode(data, None)
    if self._size == 0:
        self.head = new_node
    else:
        before_node = self.head
        while before_node._next != None:
            before_node =  before_node._next
```

**گام۱:** تعریف گره

**گام۲:** اضافه کردن گره به لیست

- دستورات لازم در صورتی که لیست خالی باشد:

- دستورات لازم در صورتی که لیست خالی نباشد:




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- **تابع درج یک گره در انتهای لیست (با داشتن داده مربوط به گره):**

```
def Add2EndByValue(self, data):
    new_node= self.SinglyLinkedListNode(data, None)
    if self._size == 0:
        self.head = new_node
    else:
        before_node = self.head
        while before_node._next != None:
            before_node =  before_node._next
        before_node._next=new_node
    self._size += 1
    return self.head
```

**گام۱:** تعریف گره

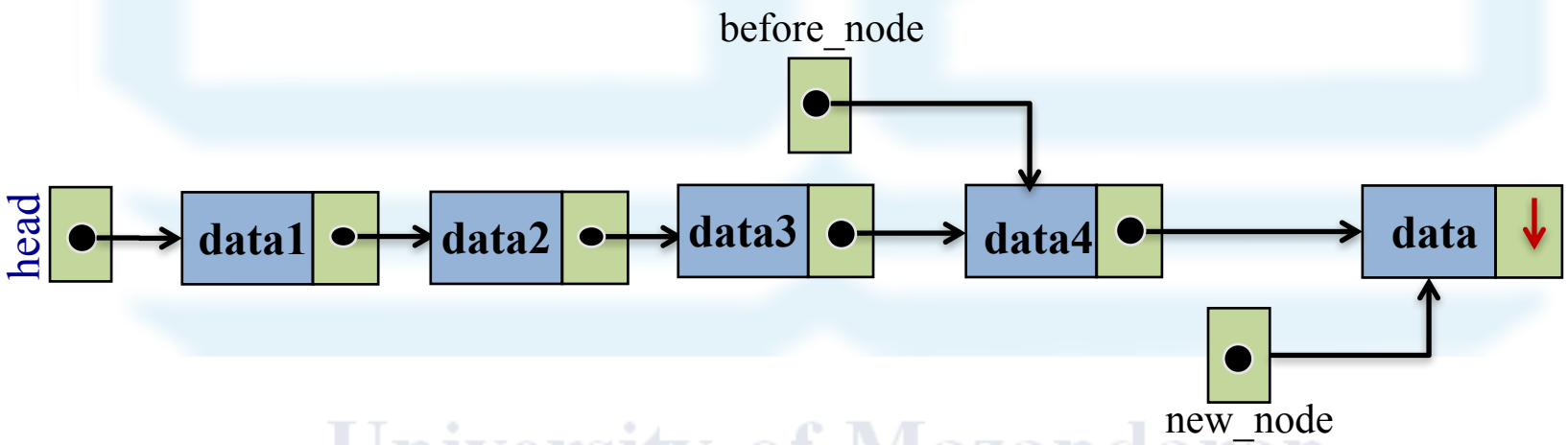
**گام۲:** اضافه کردن گره به لیست

- دستورات لازم در صورتی که لیست خالی باشد:

- دستورات لازم در صورتی که لیست خالی نباشد:

**گام۳:** اضافه کردن یک واحد به _size

**گام۴:** برگرداندن ارجاع به گره اول لیست به عنوان خروجی تابع

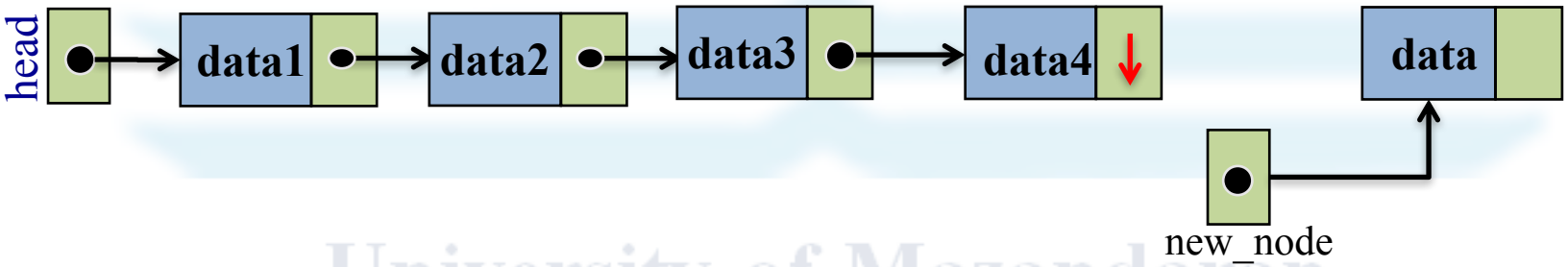



University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- تابع درج یک گره در انتهای لیست (با داشتن آدرس گره):

```
def Add2EndByReference(self, new_node):
```



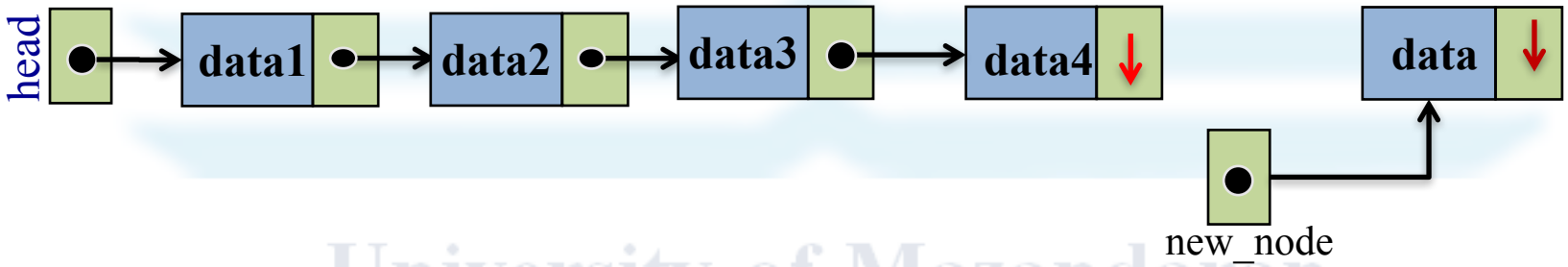

University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- **تابع درج یک گره در انتهای لیست (با داشتن آدرس گره):**

```
def Add2EndByReference(self, new_node):
    new_node._next = None
```

**گام۱:** تنظیم اشاره گر _next گره




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

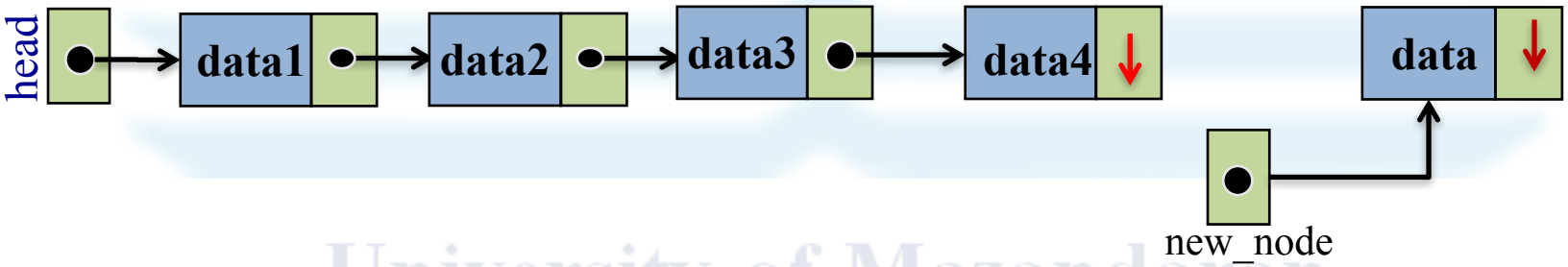
- **تابع درج یک گره در انتهای لیست (با داشتن آدرس گره):**

```
def Add2EndByReference(self, new_node):
    new_node._next = None
    if self._size == 0:
        self.head = new_node
```

**گام۱:** تنظیم اشاره گر _next گره

**گام۲:** اضافه کردن گره به لیست

- دستورات لازم در صورتی که لیست خالی باشد:




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- **تابع درج یک گره در انتهای لیست (با داشتن آدرس گره):**

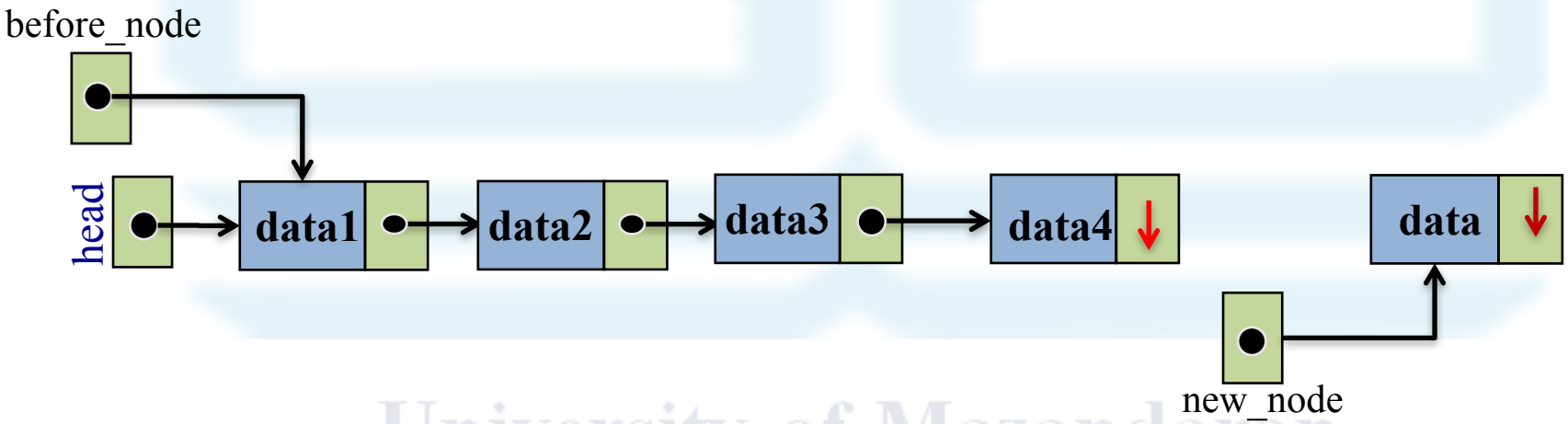
```
def Add2EndByReference(self, new_node):
    new_node._next = None

    if self._size == 0:
        self.head = new_node
    else:
        before_node = self.head
```

**گام۱:** تنظیم اشاره گر _next گره

**گام۲:** اضافه کردن گره به لیست

- دستورات لازم در صورتی که لیست خالی باشد:

- دستورات لازم در صورتی که لیست خالی نباشد:




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- **تابع درج یک گره در انتهای لیست (با داشتن آدرس گره):**

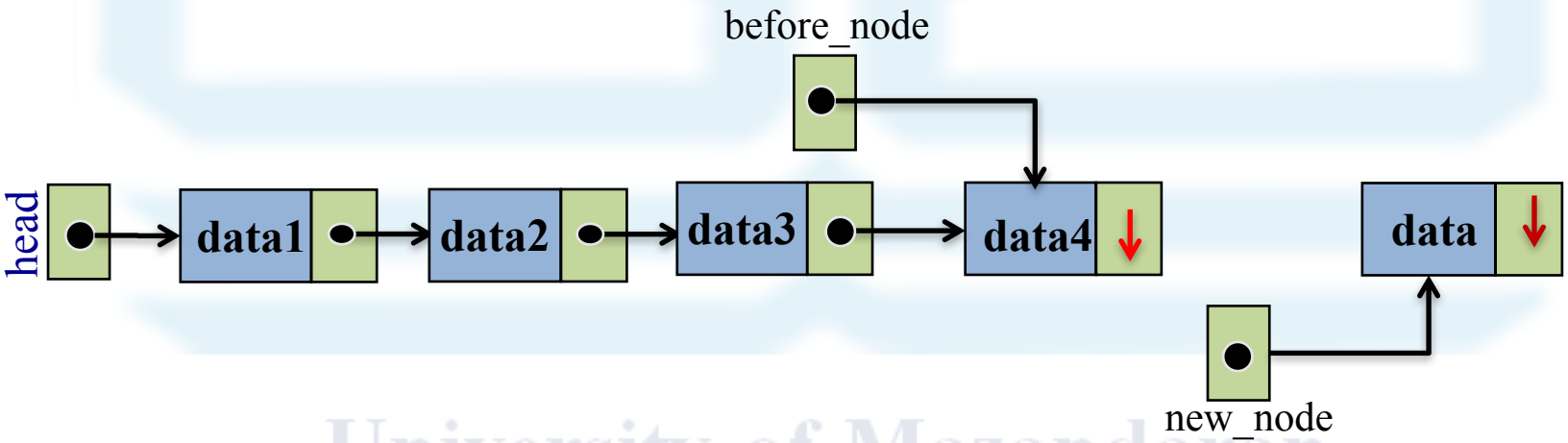
```
def Add2EndByReference(self, new_node):
    new_node._next = None
    if self._size == 0:
        self.head = new_node
    else:
        before_node = self.head
        while before_node._next != None:
            before_node =  before_node._next
```

**گام۱:** تنظیم اشاره گر _next گره

**گام۲:** اضافه کردن گره به لیست

- دستورات لازم در صورتی که لیست خالی باشد:

- دستورات لازم در صورتی که لیست خالی نباشد:




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- **تابع درج یک گره در انتهای لیست (با داشتن آدرس گره):**

```
def Add2EndByReference(self, new_node):
    new_node._next = None
    if self._size == 0:
        self.head = new_node
    else:
        before_node = self.head
        while before_node._next != None:
            before_node =  before_node._next
        before_node._next=new_node
    self._size += 1
    return self.head
```

**گام۱:** تنظیم اشاره گر _next گره

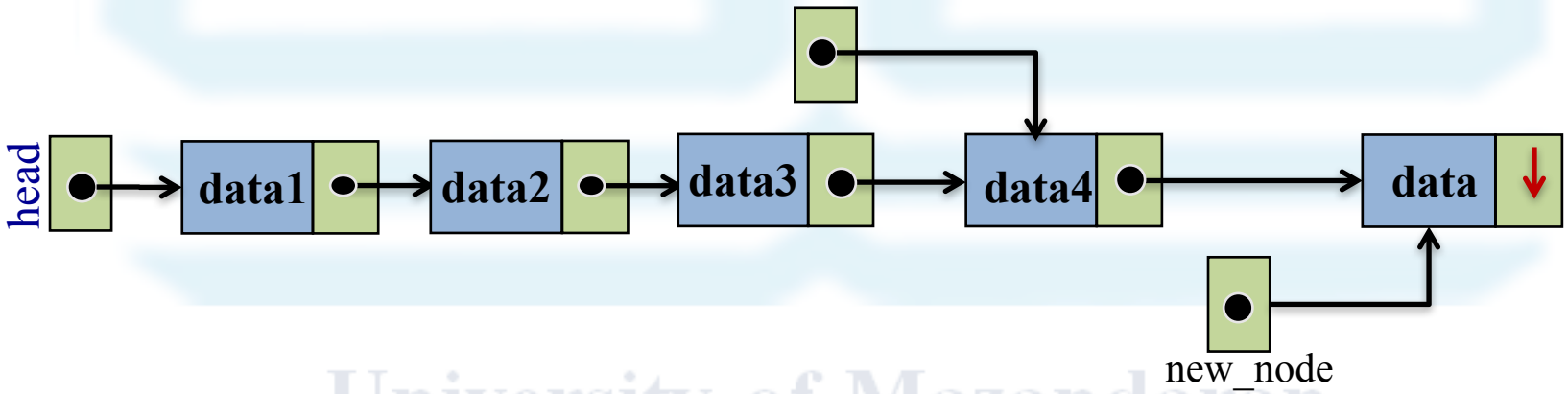
**گام۲:** اضافه کردن گره به لیست

- دستورات لازم در صورتی که لیست خالی باشد:

- دستورات لازم در صورتی که لیست خالی نباشد:

**گام۳:** اضافه کردن یک واحد به _size

**گام۴:** برگرداندن ارجاع به گره اول لیست به عنوان خروجی تابع




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

قطعه برنامه بهینه‌تر

- **تابع درج یک گره در انتهای لیست:**

```
def Add2End (self, inp):
    if type(self.head) != type(inp):
        new_node= self.SinglyLinkedListNode(inp, None)
    else:
        new_node=inp
        new_node._next = None
    if self._size == 0:
        self.head = new_node
    else:
        before_node = self.head
        while before_node._next != None:
            before_node =  before_node._next
        before_node._next=new_node
    self._size += 1
    return self.head
```

**گام۱:** تنظیم اولیه:

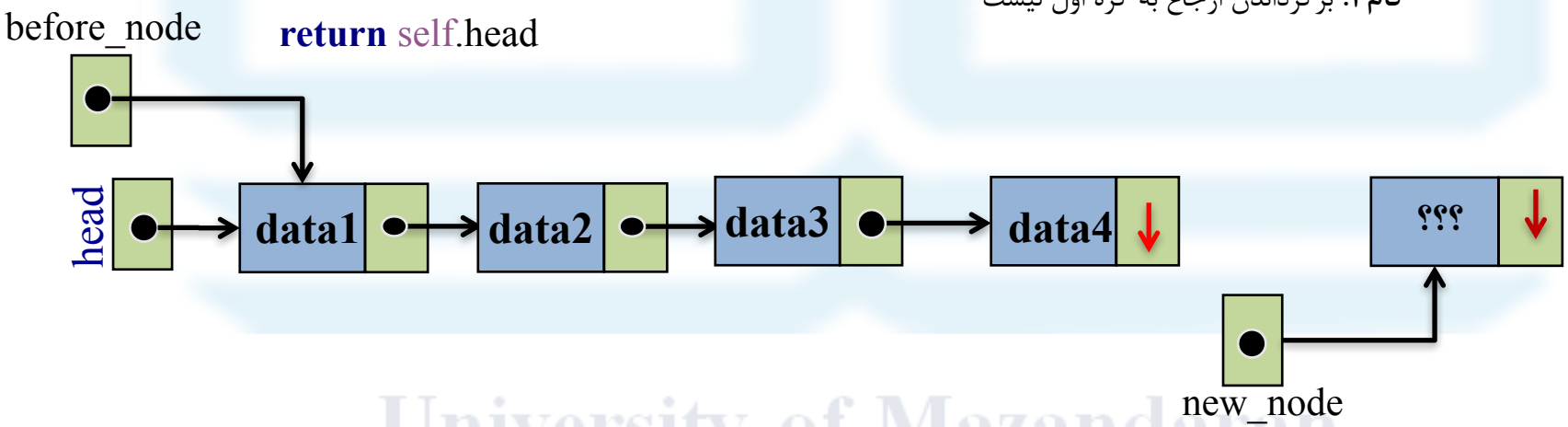
**گام۲:** اضافه کردن گره به لیست

- دستورات لازم در صورتی که لیست خالی باشد:

- دستورات لازم در صورتی که لیست خالی نباشد:

**گام۳:** اضافه کردن یک واحد به _size

**گام۴:** برگرداندن ارجاع به گره اول لیست




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

قطعه برنامه بهینه تر

- **تابع درج یک گره در انتهای لیست:**

```
def Add2End (self, inp):
    if type(self.head) != type(inp):
        new_node= self.SinglyLinkedListNode(inp, None)
    else:
        new_node=inp
        new_node._next = None
    if self._size == 0:
        self.head = new_node
    else:
        before_node = self.head
        while before_node._next != None:
            before_node =  before_node._next
        before_node._next=new_node
    self._size += 1
    return self.head
```

**گام۱:** تنظیم اولیه:

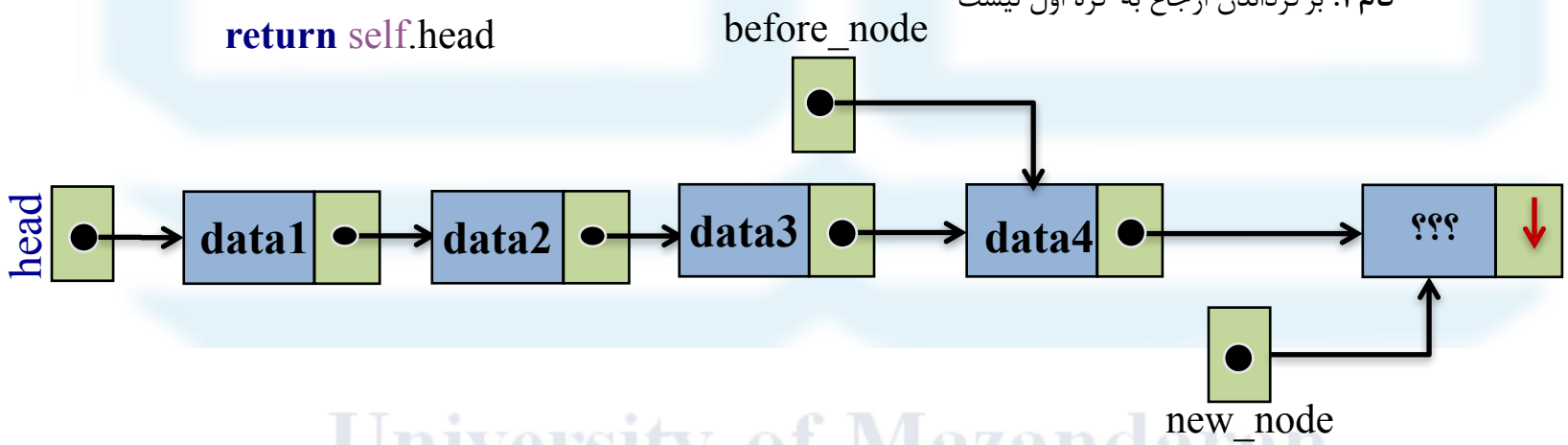
**گام۲:** اضافه کردن گره به لیست

- دستورات لازم در صورتی که لیست خالی باشد:

- دستورات لازم در صورتی که لیست خالی نباشد:

**گام۳:** اضافه کردن یک واحد به _size

**گام۴:** برگرداندن ارجاع به گره اول لیست

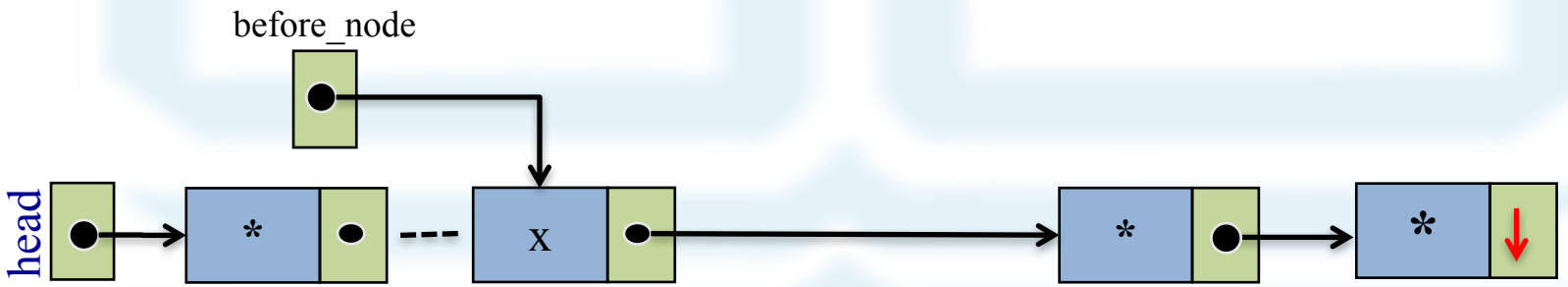



University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- تابع درج یک گره با مقدار داده‌ای y، بعد از گره‌ای با مقدار داده‌ای x (در صورت وجود):

```
def AddAfterNodeByValue(self, x , y):
    before_node = SearchNodeByValue(self,x):
```

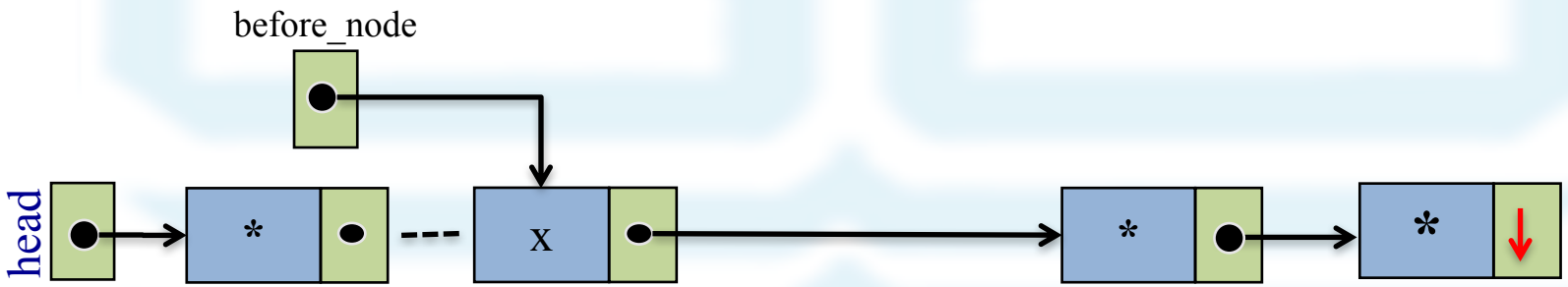



University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- تابع درج یک گره با مقدار داده‌ای y، بعد از گره‌ای با مقدار داده‌ای x (در صورت وجود):

```
def AddAfterNodeByValue(self, x , y):
    before_node = SearchNodeByValue(self,x):
    if before_node == None :
        return 0
    else:
        ????
```

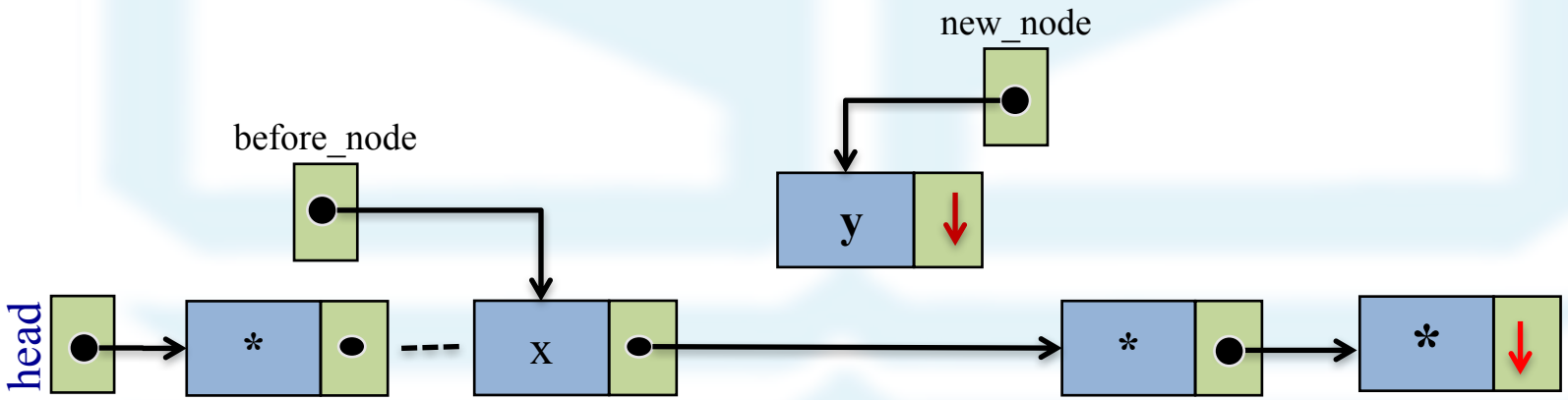



University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- تابع درج یک گره با مقدار داده‌ای y، بعد از گره‌ای با مقدار داده‌ای x (در صورت وجود):

```
def AddAfterNodeByValue(self, x , y):
    before_node = SearchNodeByValue(self,x):
    if before_node == None :
        return 0
    else:
        new_node= self.SinglyLinkedListNode(y, None)
```

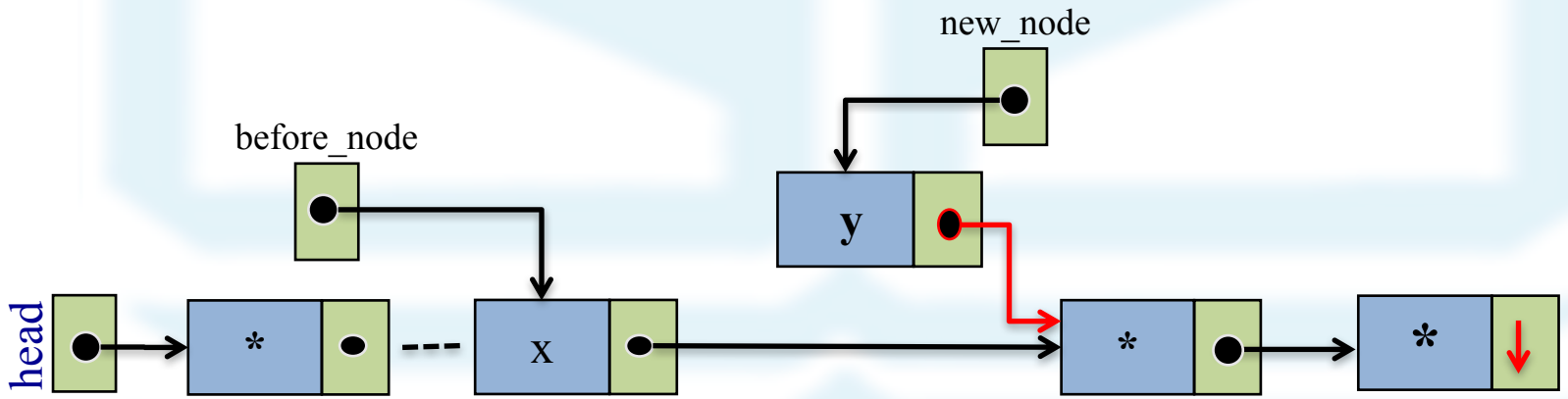



University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- تابع درج یک گره با مقدار داده‌ای y، بعد از گره‌ای با مقدار داده‌ای x (در صورت وجود):

```
def AddAfterNodeByValue(self, x , y):
    before_node = SearchNodeByValue(self,x):
    if before_node == None :
        return 0
    else:
        new_node= self.SinglyLinkedListNode(y, None)
        new_node._next=before_node._next
```

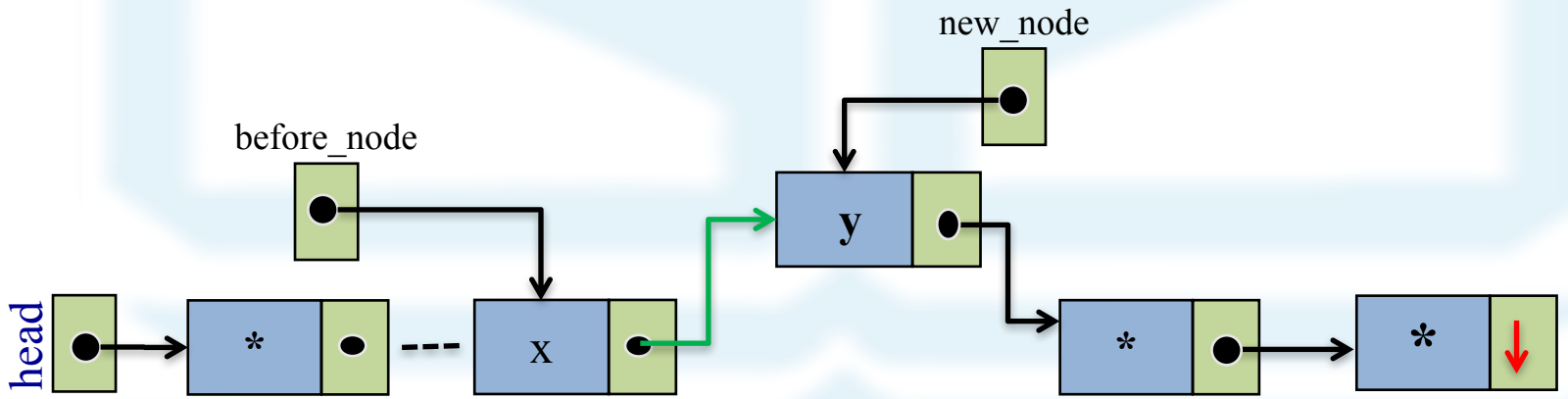



University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- تابع درج یک گره با مقدار داده‌ای y، بعد از گره‌ای با مقدار داده‌ای x (در صورت وجود):

```
def AddAfterNodeByValue(self, x , y):
    before_node = SearchNodeByValue(self,x):
    if before_node == None :
        return 0
    else:
        new_node= self.SinglyLinkedListNode(y, None)
        new_node._next=before_node._next
        before_node._next=new_node
```



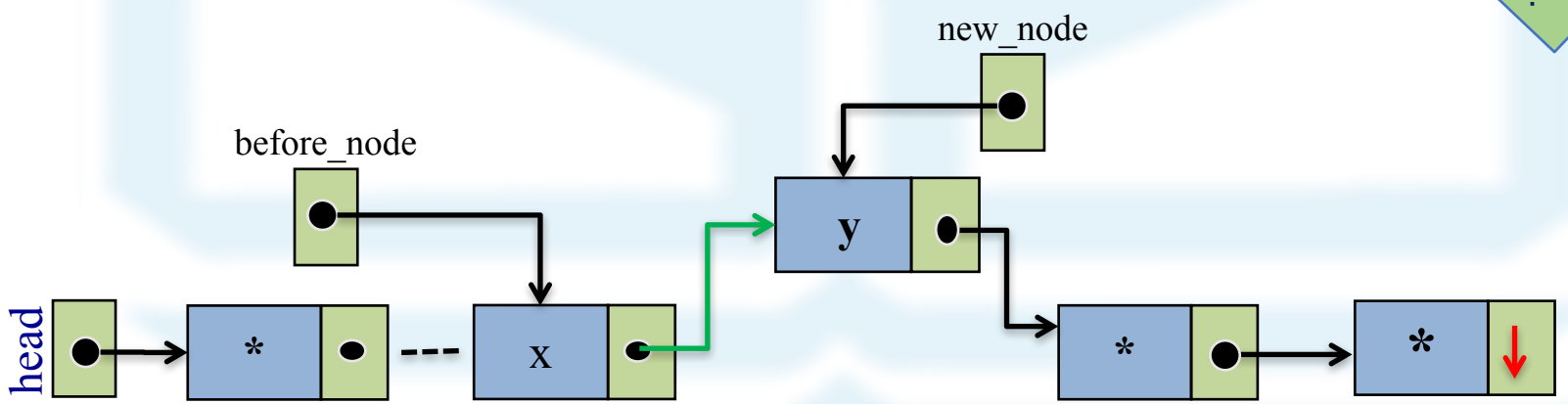

University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- تابع درج یک گره با مقدار داده‌ای y، بعد از گره‌ای با مقدار داده‌ای x (در صورت وجود):

```
def AddAfterNodeByValue(self, x , y):
    before_node = SearchNodeByValue(self,x):
    if before_node == None :
        return 0
    else:
        new_node= self.SinglyLinkedListNode(y, None)
        new_node._next=before_node._next
        before_node._next=new_node
        self._size += 1
    return 1
```

اگر فقط آخرین گره لیست دارای مقدار داده‌ای x باشد، آیا این قطعه برنامه درست عمل می‌کند. چرا؟




University of Mazandaran

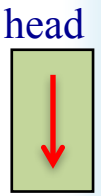
# پیاده‌سازی لیست تک پیوندی

- **تابع حذف یک گره از ابتدای لیست:**

**گام۱:** دستورات حذف یک گره:

- دستورات لازم در صورتی که لیست خالی باشد:

```
def RemoveFirstNode (self):
    if self._size == 0:
        return 0, None
```




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

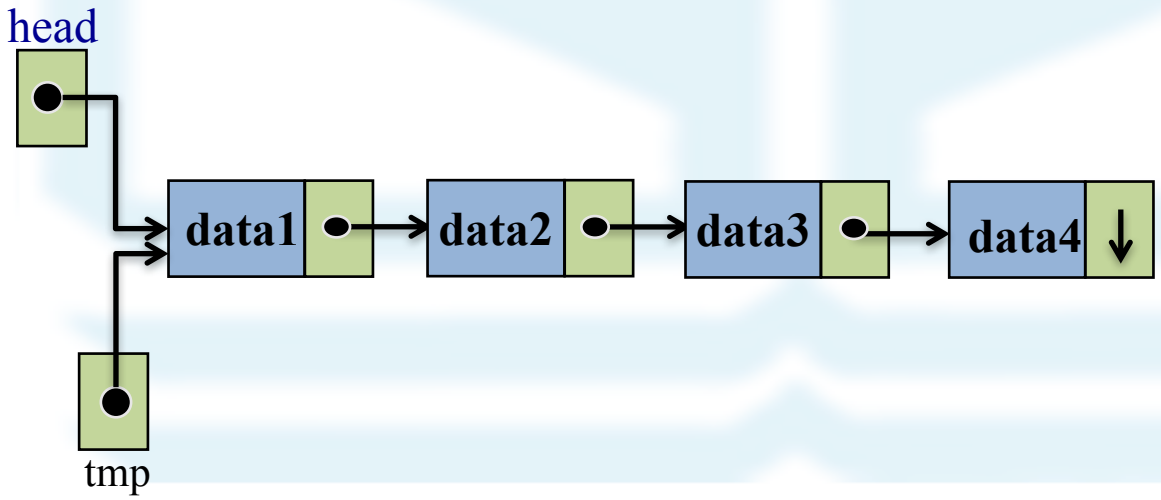
- **تابع حذف یک گره از ابتدای لیست:**

```
def RemoveFirstNode (self):
    if self._size == 0:
        return 0, None
    else:
        tmp = self.head
```

**گام۱:** دستورات حذف یک گره:

- دستورات لازم در صورتی که لیست خالی باشد:

- دستورات لازم در صورتی که لیست خالی نباشد:




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

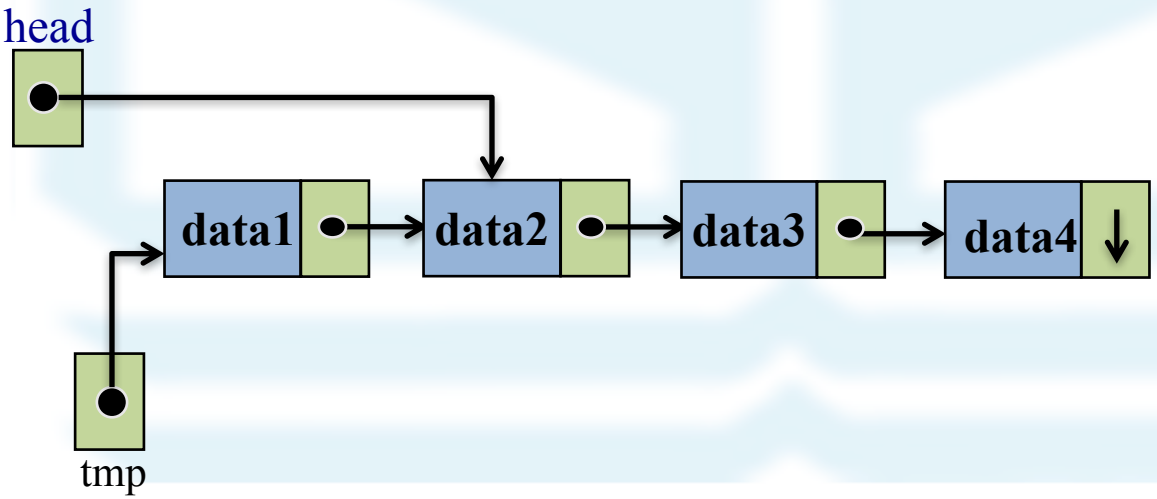
- **تابع حذف یک گره از ابتدای لیست:**

**گام۱:** دستورات حذف یک گره:

- دستورات لازم در صورتی که لیست خالی باشد:

- دستورات لازم در صورتی که لیست خالی نباشد:

```
def RemoveFirstNode (self):
    if self._size == 0:
        return 0, None
    else:
        tmp = self.head
        self.head = self.head._next
```




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- **تابع حذف یک گره از ابتدای لیست:**

```
def RemoveFirstNode (self):
    if self._size == 0:
        return 0, None
    else:
        tmp = self.head
        self.head = self.head._next
        self._size -= 1
        return 1, tmp
```

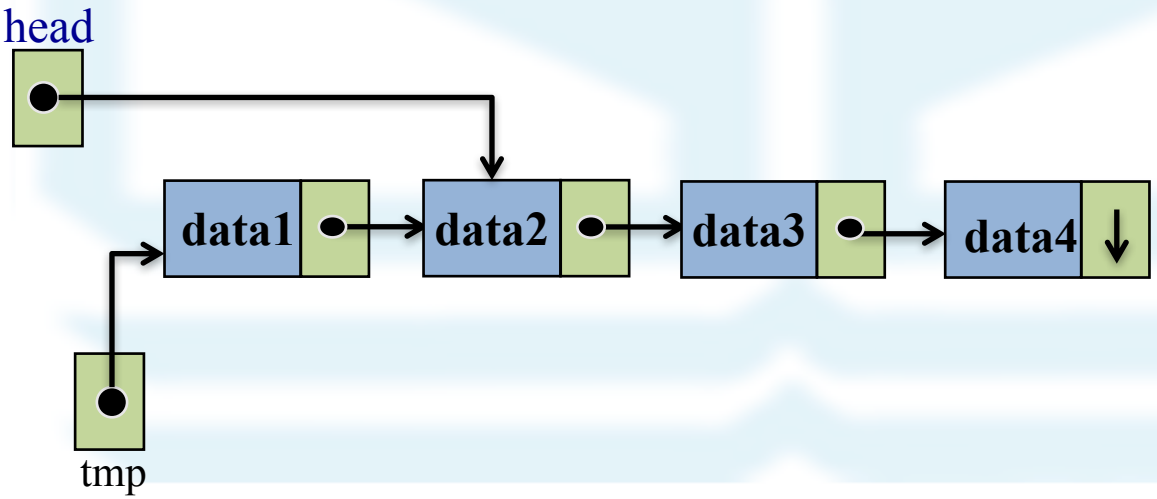
**گام۱:** دستورات حذف یک گره:

- دستورات لازم در صورتی که لیست خالی باشد:

- دستورات لازم در صورتی که لیست خالی نباشد:

**گام۳:** کم کردن یک واحد از _size

**گام۴:** برگرداندن ارجاع به گره حذف شده به عنوان خروجی تابع




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- **تابع حذف یک گره از انتهای لیست:**

```
def RemoveLastNode (self):
    if self._size == 0:
        return 0, None
    elif self._size ==1:
        flag, delete_node = self.RemoveFirstNode ():
    else:
        before_node = self.head
        while before_node._next._next != None:
            before_node =  before_node._next
        delete_node = before_node._next
        before_node._next = None
        flag = 1
        self._size -= 1
    return flag, delete_node
```

**گام۱:** دستورات حذف یک گره:

- دستورات لازم در صورتی که لیست خالی باشد:

- دستورات لازم در صورتی که لیست فقط شامل یک گره باشد:

- دستورات لازم در صورتی که لیست شامل بیش از یک گره باشد:

**گام۳:** کم کردن یک واحد از _size

**گام۴:** برگرداندن ارجاع به گره حذف شده به عنوان خروجی تابع

University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- تابع حذف گره‌ای با مقدار داده‌ای x از لیست(در صورت وجود):

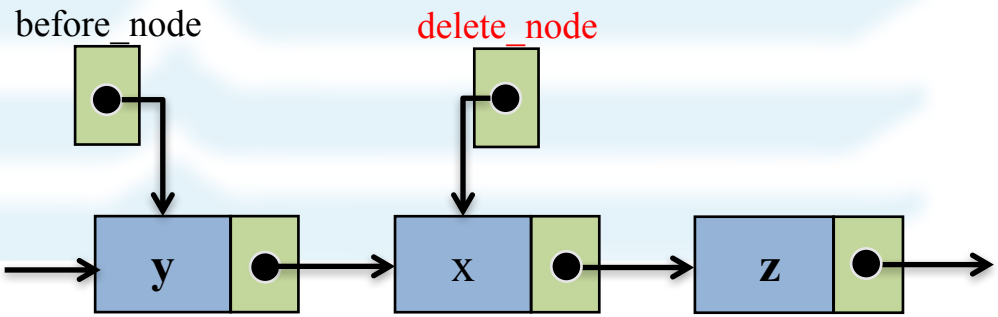
```
def RemoveNodeByValue (self, x):
    delete_node = SearchNodeByValue(self, x):
    if delete_node == None:
        return 0, None
    else:
        if self._size ==1 :
            flag, delete_node = self.RemoveFirstNode ():
        elif delete_node._next == None:
            flag, delete_node = self.RemoveLastNode ():
```

University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- تابع حذف گره‌ای با مقدار داده‌ای x از لیست(در صورت وجود):

```
def RemoveNodeByValue (self, x):
    delete_node = SearchNodeByValue(self, x):
    if delete_node == None:
        return 0, None
    else:
        if self._size ==1 :
            flag, delete_node = self.RemoveFirstNode ():
        elif delete_node._next == None:
            flag, delete_node = self.RemoveLastNode ():
        else:
            before_node = self.head
            while before_node._next != delete_node :
                before_node =  before_node._next
```

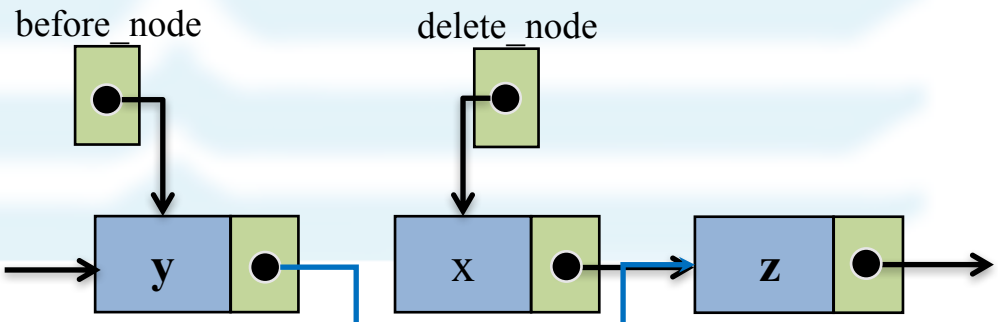



University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

- تابع حذف گره‌ای با مقدار داده‌ای x از لیست(در صورت وجود):

```
def RemoveNodeByValue (self, x):
    delete_node = SearchNodeByValue(self, x):
    if delete_node == None:
        return 0, None
    else:
        if self._size ==1 :
            flag, delete_node = self.RemoveFirstNode ():
        elif delete_node._next == None:
            flag, delete_node = self.RemoveLastNode ():
        else:
            before_node = self.head
            while before_node._next != delete_node :
                before_node =  before_node._next
            before_node._next = delete_node._next
```

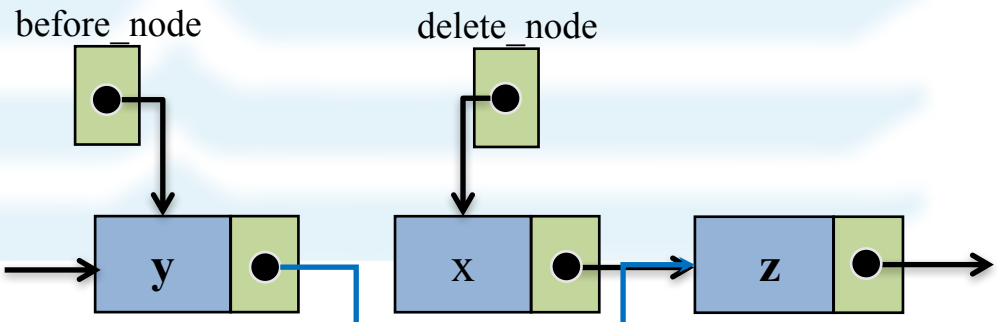



University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

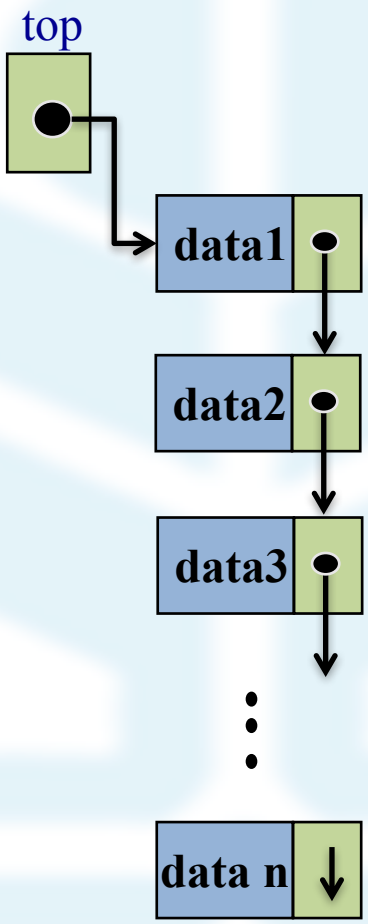
- تابع حذف گره‌ای با مقدار داده‌ای x از لیست(در صورت وجود):

```
def RemoveNodeByValue (self, x):
    delete_node = SearchNodeByValue(self, x):
    if delete_node == None:
        return 0, None
    else:
        if self._size ==1 :
            flag, delete_node = self.RemoveFirstNode ():
        elif delete_node._next == None:
            flag, delete_node = self.RemoveLastNode ():
        else:
            before_node = self.head
            while before_node._next != delete_node :
                before_node =  before_node._next
            before_node._next = delete_node._next
            flag = 1
            self._size -= 1
        return flag, delete_node
```



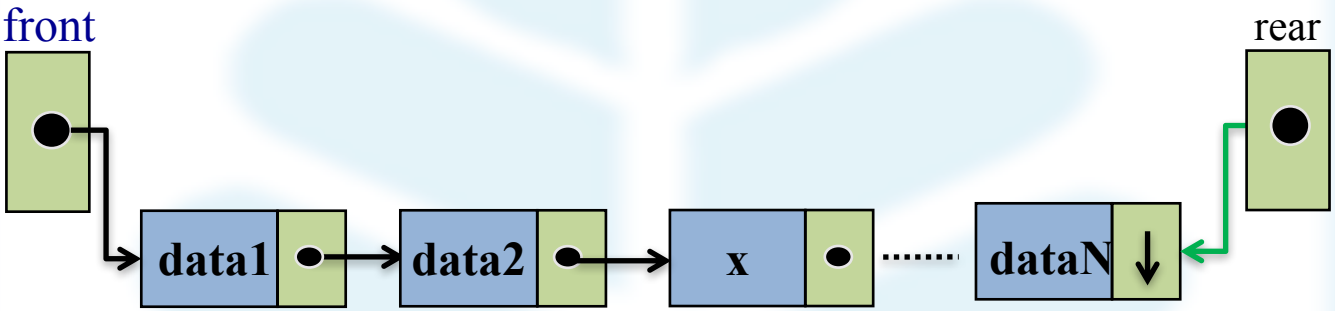

University of Mazandaran

# پیاده‌سازی پشته با لیست تک پیوندی



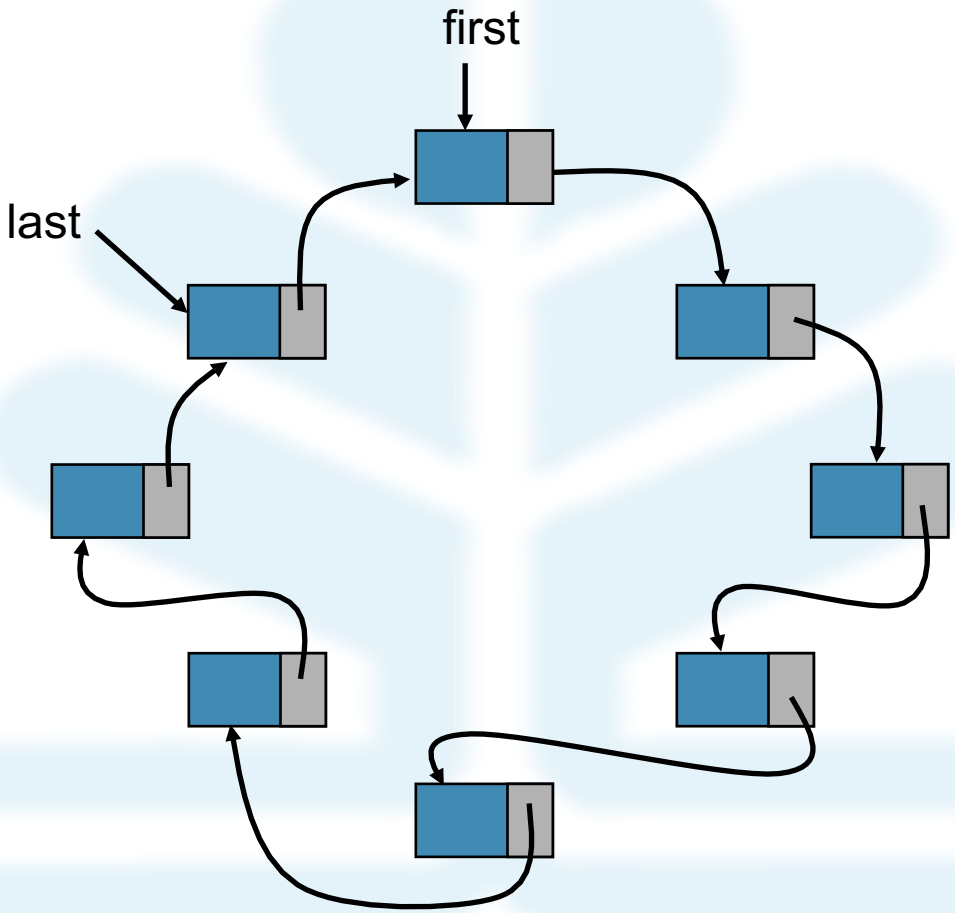

University of Mazandaran

# پیاده‌سازی صف با لیست تک پیوندی




University of Mazandaran

# پیاده‌سازی صف حلقوی با لیست تک پیوندی




University of Mazandaran

# پیاده‌سازی لیست تک پیوندی

## تمرین

۱- تابعی بنویسید که گره‌ای با مقدار داده‌ای x را جستجو نماید و در صورت وجود، آدرس گره قبلی آن را برگرداند.

۲- تابعی بنویسید که گره‌ای با آدرس معلوم را جستجو نماید و در صورت وجود، آدرس گره قبلی آن را برگرداند.

۳- تابعی بنویسید که گره‌ای با مقدار داده‌ای x را جستجو نماید و در صورت وجود، آدرس گره بعدی آن را برگرداند.

۴- تابعی بنویسید که گره‌ای با آدرس معلوم را جستجو نماید و در صورت وجود، آدرس گره بعدی آن را برگرداند.

۵- تابعی بنویسید که گره‌ای با آدرس معلوم را قبل از گره‌ای با مقدار داده‌ای x (در صورت وجود) درج نماید.

۶- تابعی بنویسید که گره‌ای با آدرس معلوم را بعد از گره‌ای با مقدار داده‌ای x (در صورت وجود) درج نماید.

۷- تابعی بنویسید که گره‌ای با مقدار داده‌ای y را قبل از گره‌ای با مقدار داده‌ای x (در صورت وجود) درج نماید.

۸- تابعی بنویسید که گره‌ای با آدرسی خاص را در صورت وجود از لیست حذف کند.

* برنامه کاملی بنویسید که از لیست تک پیوندی برای پیاده‌سازی پشته استفاده نماید.

** برنامه کاملی بنویسید که از لیست تک پیوندی برای پیاده‌سازی صف استفاده نماید.

*** برنامه کاملی بنویسید که از لیست تک پیوندی برای پیاده‌سازی صف حلقوی استفاده نماید.

**** برنامه کاملی بنویسید که از لیست تک پیوندی برای پیاده‌سازی مساله سیم کشی استفاده نماید.

University of Mazandaran