ساختمان داده ها

نوع داده مجردِ لیست خطی

مدرس: غیاثی‌شیرازی
دانشگاه فردوسی مشهد

Linear List

- لیست خطی، یک نوع داده مجرد است که می تواند اشیاء داده ای را در یک لیست ذخیره کند.

۱ ، 2 ، 7، 4، 5

"Ali", "Farvadin", ...

عملیات پایه ای لیست خطی مبتنی بر اندیس

- درج عنصری در لیست (در محل مشخص شده با اندیس).
- حذف عنصری از لیست (از محل مشخص شده با اندیس).
- گرفتن عنصری از لیست (با محل مشخص شده با اندیس).

۱, ۷, ۴, ۳, 2

- عملیاتی که در لیست تغییر ایجاد می کنند می توانند منجر به نامعتبر شدن پیمایشگرها شوند.

insert(3, 0) ⇒ ۱, ۷, ۳, ۴, ۳, 2 (محل)

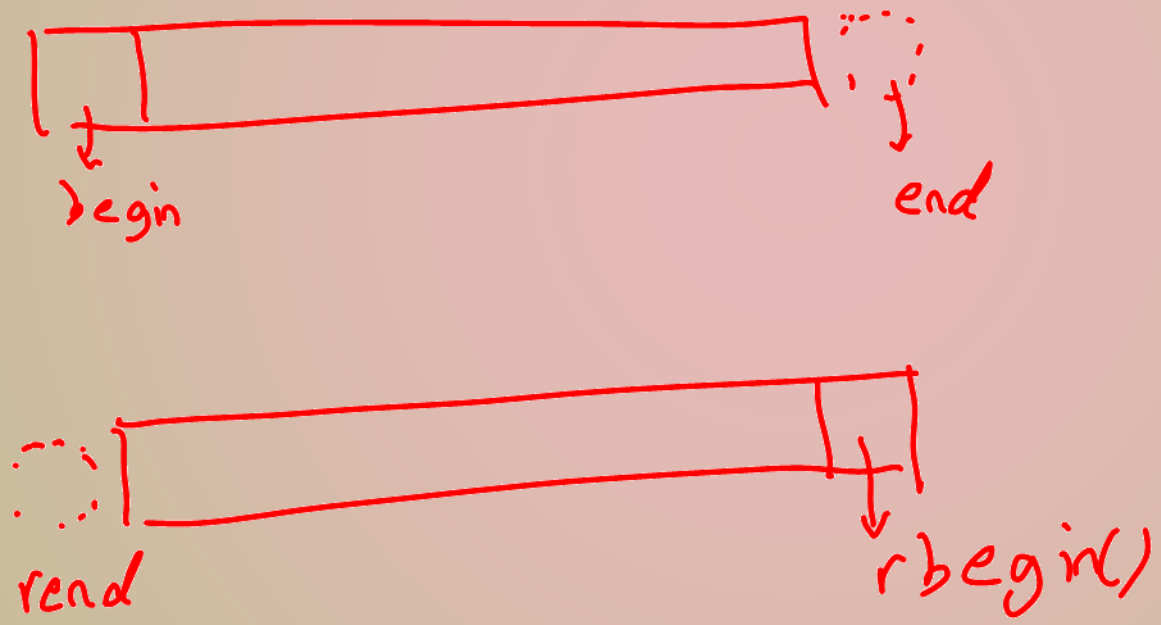
remove(3) ⇒ ۱, ۷, ۳, 2

get(3) ⇒ ۴

با رفتن اندیس دهم از ۱

عملیات پیمایش

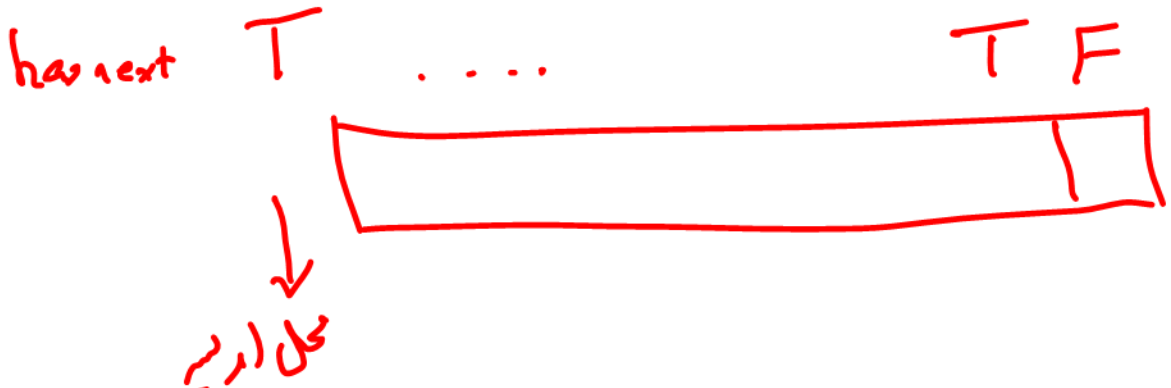
- با حمایت از عملیات زیر، پیمایش لیست خطی در هر دو جهت ممکن خواهد بود.



- begin() –
- end() –
- rbegin() –
- rend() –

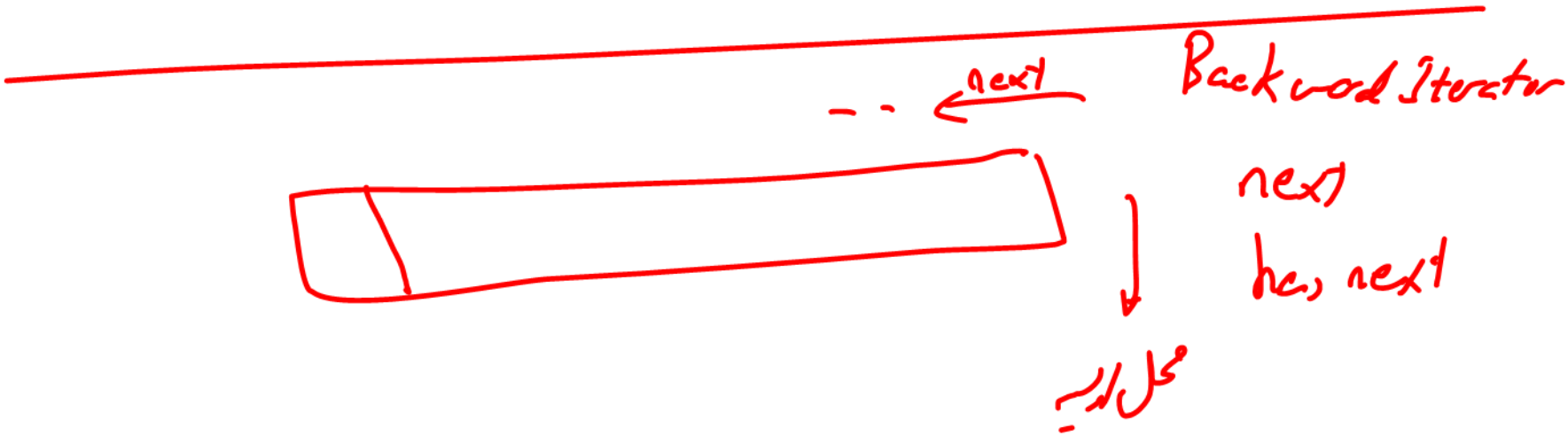
در جاوا

Forward Iterator



next

has next

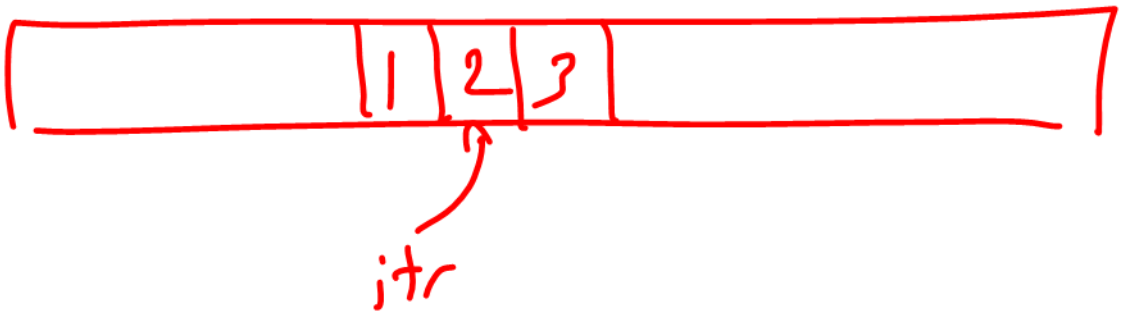


عملیات پایه ای با استفاده از پیمایشگرها

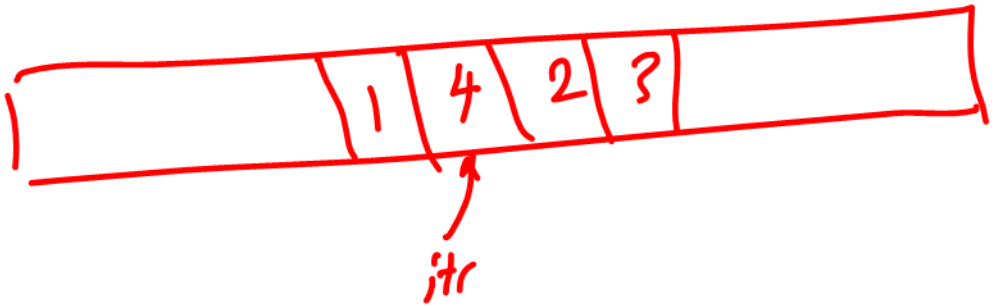
- درج عنصری در لیست (در محل مشخص شده با پیمایشگر).
- حذف عنصری از لیست (از محل مشخص شده با پیمایشگر).
  - پس از حذف، پیمایشگری که به عنصر بعد از عنصر حذف شده اشاره می کند برگردانده می شود.

- اگر اجازه دهیم که پیمایشگر از نوع دسترسی تصادفی باشد، آنگاه نیازی به انجام عملیات بر اساس اندیس نیست.
  - مثلا با پیمایشگر begin()+6 می توان به عنصر هفتم لیست دسترسی پیدا کرد.

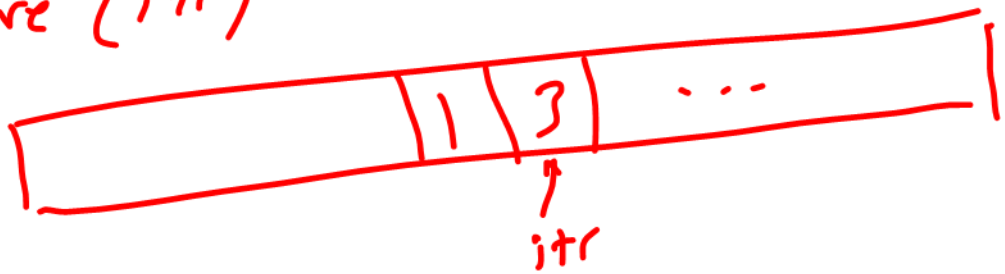
C++



insert(itr, 4)



remove (itr)

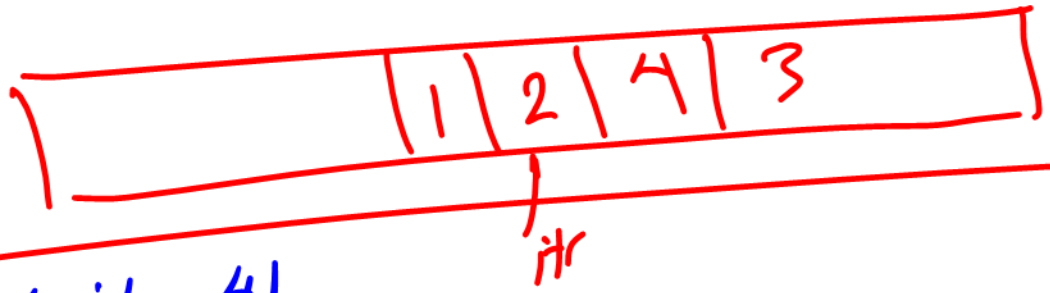


insert in Java

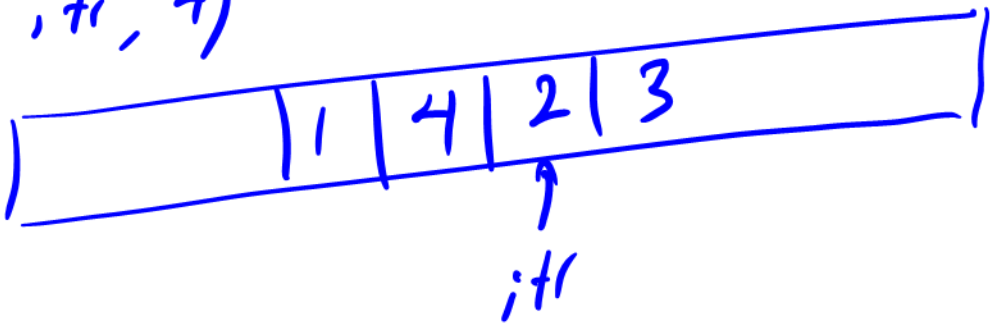
Forward Iterator

Backward Iterator

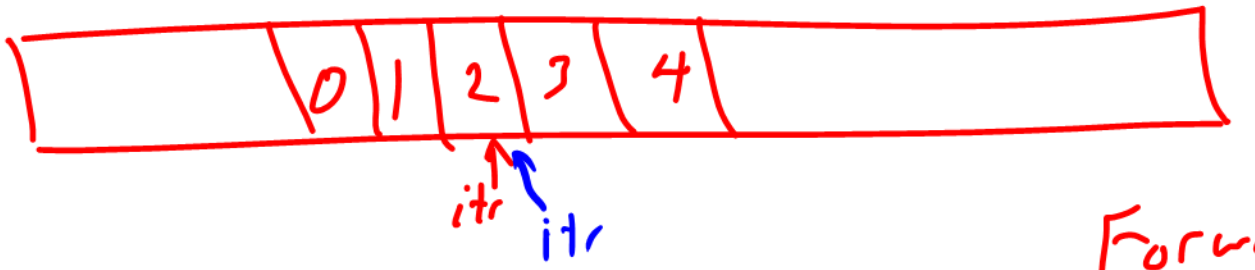
insert(itr, 4)



insert(itr, 4)



remove in Java

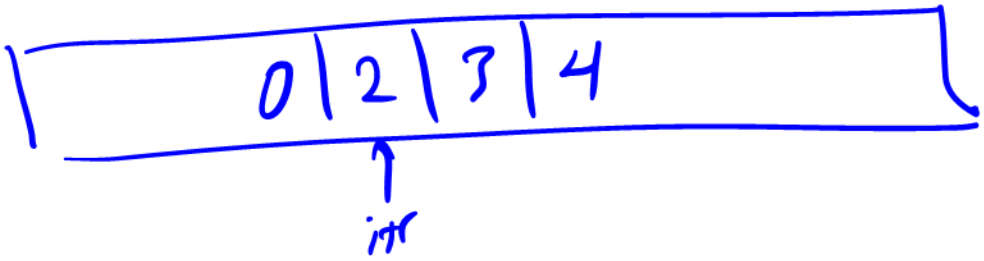


ForwardIterator

remove (itr)

BackwardIterator

remove (itr)



Java : next() hasNext()

عملیات قابل انجام بر روی پیمایشگرها

operator++() // preincrement

operator++(int) // postincrement

operator--() // predecrement

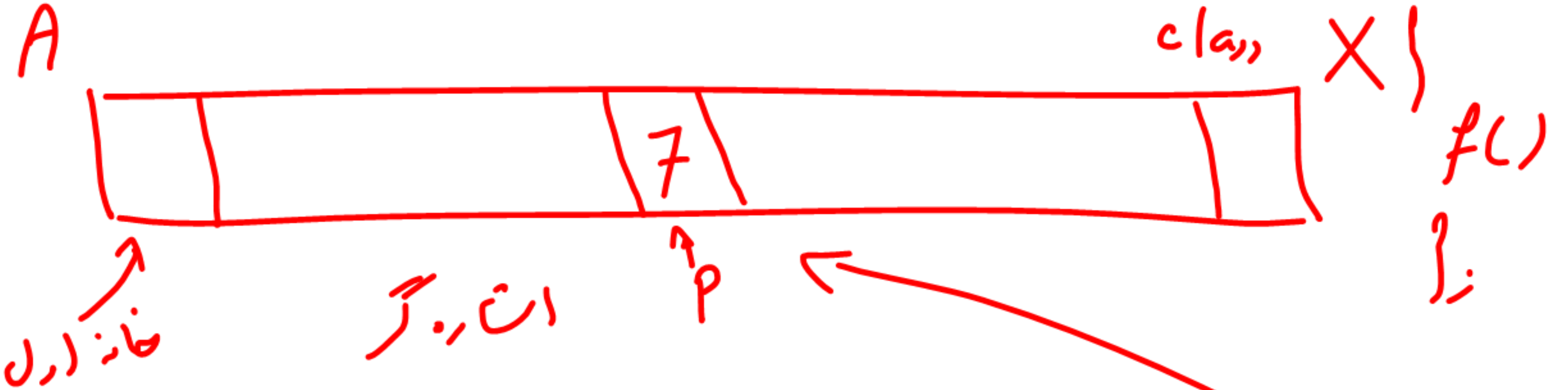
operator--(int) // postdecrement

bool operator!=(const Iterator right)const

bool operator==(const Iterator right)const

T* operator->() const

T& operator*() const



itr++

⋮

(*itr)

$q = p{+}{+}$

$q = {+}{+}p$

$q = p{-}{-}$

$q = {-}{-}p$

$p \to f()$

$(*p) =$ شیء ذخیره شده در حافظه p

$*p = 7$

عملیات اطلاع دهی

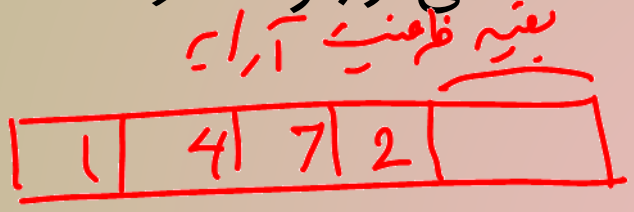
- آیا لیست خالی است؟ (isEmpty())
- چند عنصر در لیست درج شده است؟(size())

عملیات تکمیلی

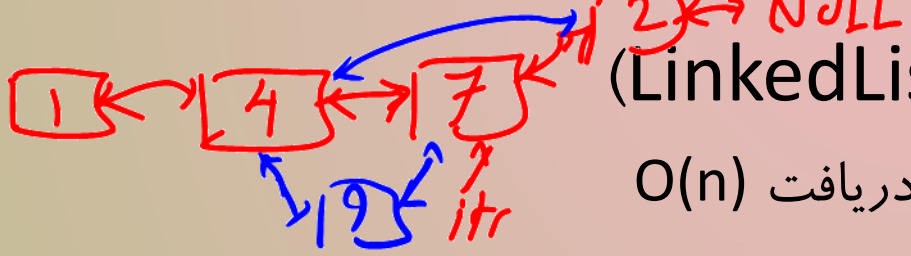
- مساوی قرار دادن لیستی با لیست دیگر
- مرتب سازی (توسط کلاس مقایسه ارسال شده به تابع)
- برعکس کردن لیست.

روش های پیاده سازی لیست خطی

- حداقل سه روش برای پیاده سازی لیست خطی وجود دارد که دو مورد از آنها عبارتند از:



1. به صورت آرایه (ArrayList)
   - زمان درج و حذف O(n) و زمان دریافت O(1)

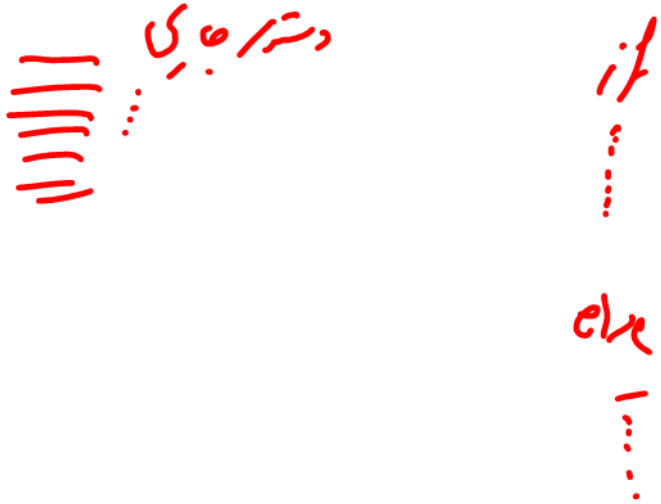


2. به صورت لیست پیوندی (LinkedList)
   - زمان درج و حذف O(1) و زمان دریافت O(n)

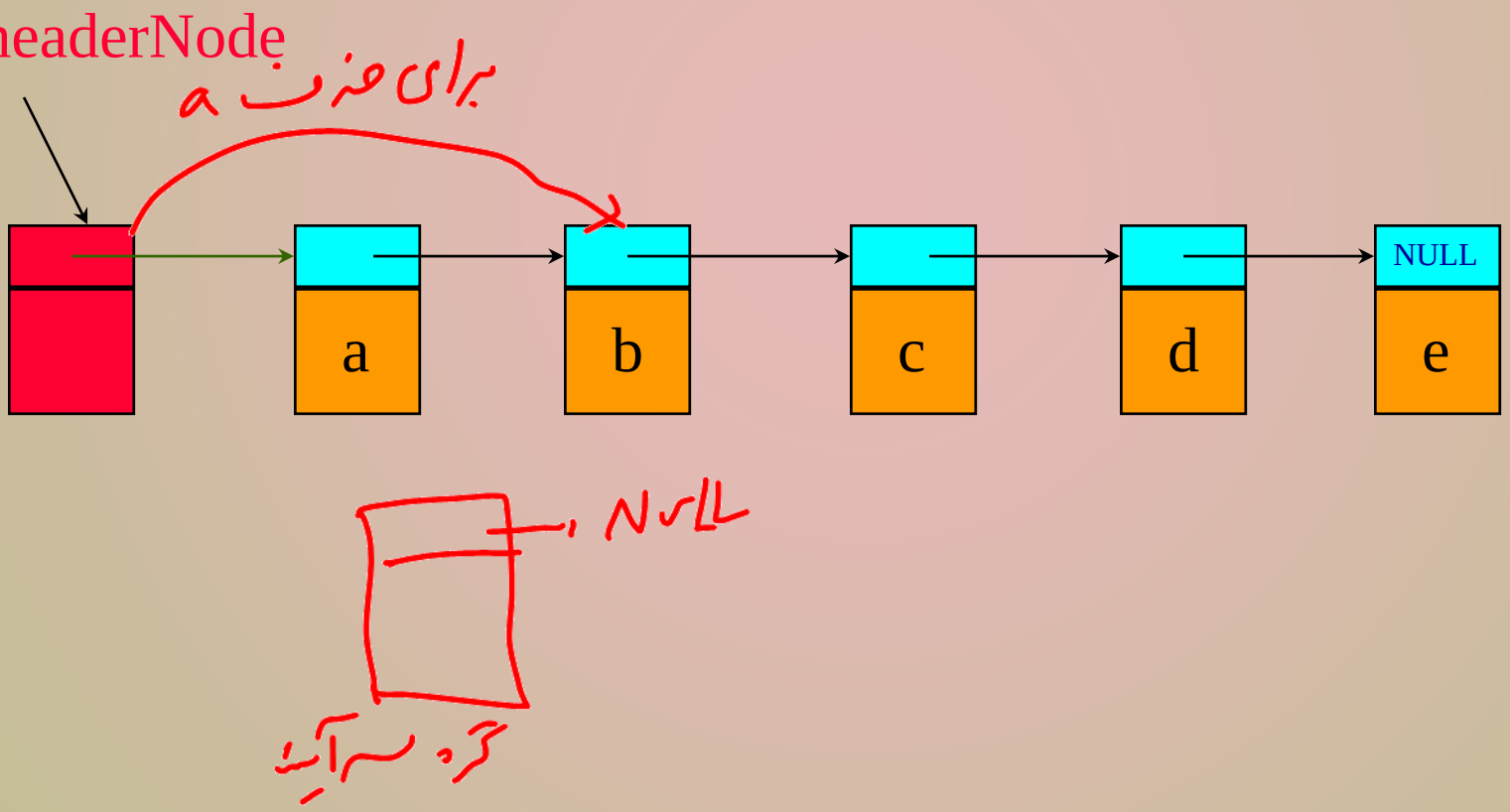
- در بخش های بعدی درس روشی برای پیاده سازی لیست خطی خواهیم دید که همه عملیات درج، حذف و دریافت را در زمان O(log n) انجام می دهد.

انواع روش های پیاده سازی لیست خطی به صورت لیست پیوندی

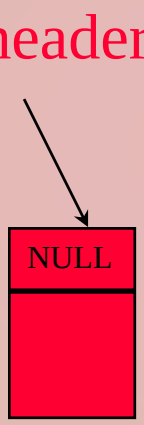
- با گره سرآیند (Header Node)
  - سبب ساده تر شدن پیاده سازی می شود (زیرا قبل از هر گرهی، گرهی وجود دارد).
- به صورت دوری (Circular)
  - امکان دور زدن از انتهای لیست به ابتدای لیست را می دهد.
- به صورت دوطرفه (Doubly Linked List)
  - دسترسی به عنصر قبلی را تسریع می بخشد.

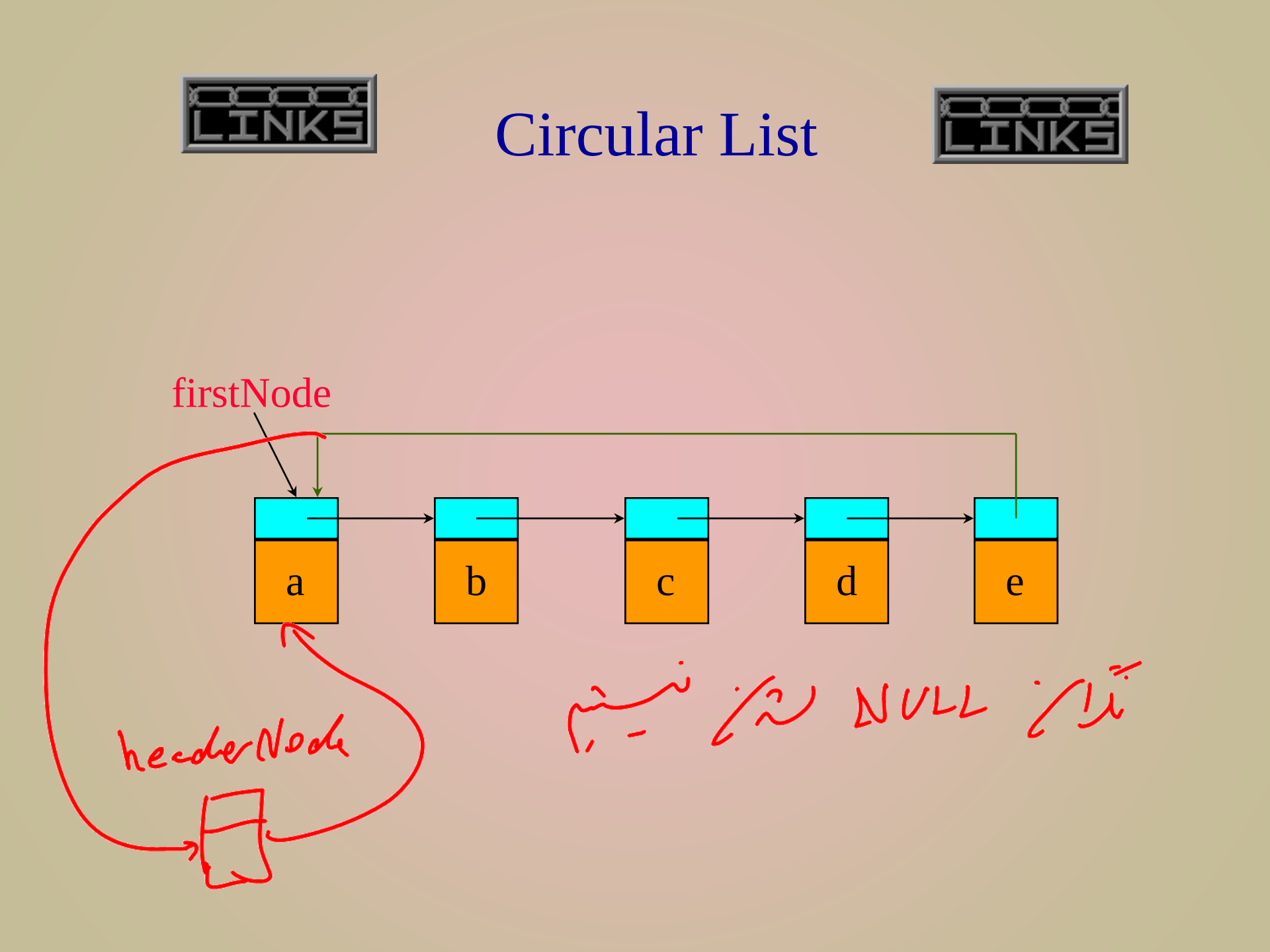

دسترسی
از
مدام

Chain With Header Node

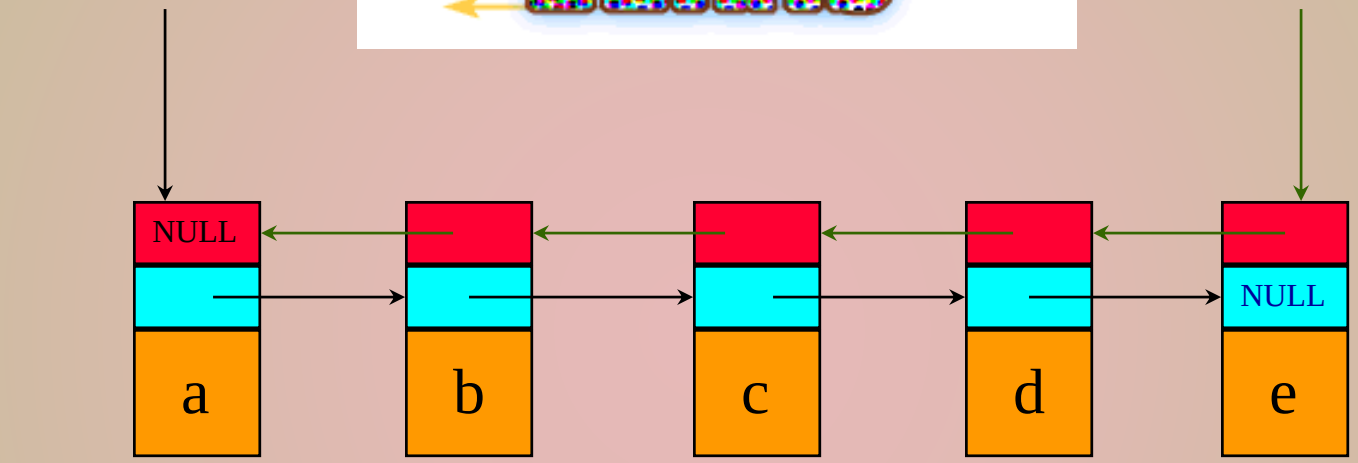


Empty Chain With Header Node

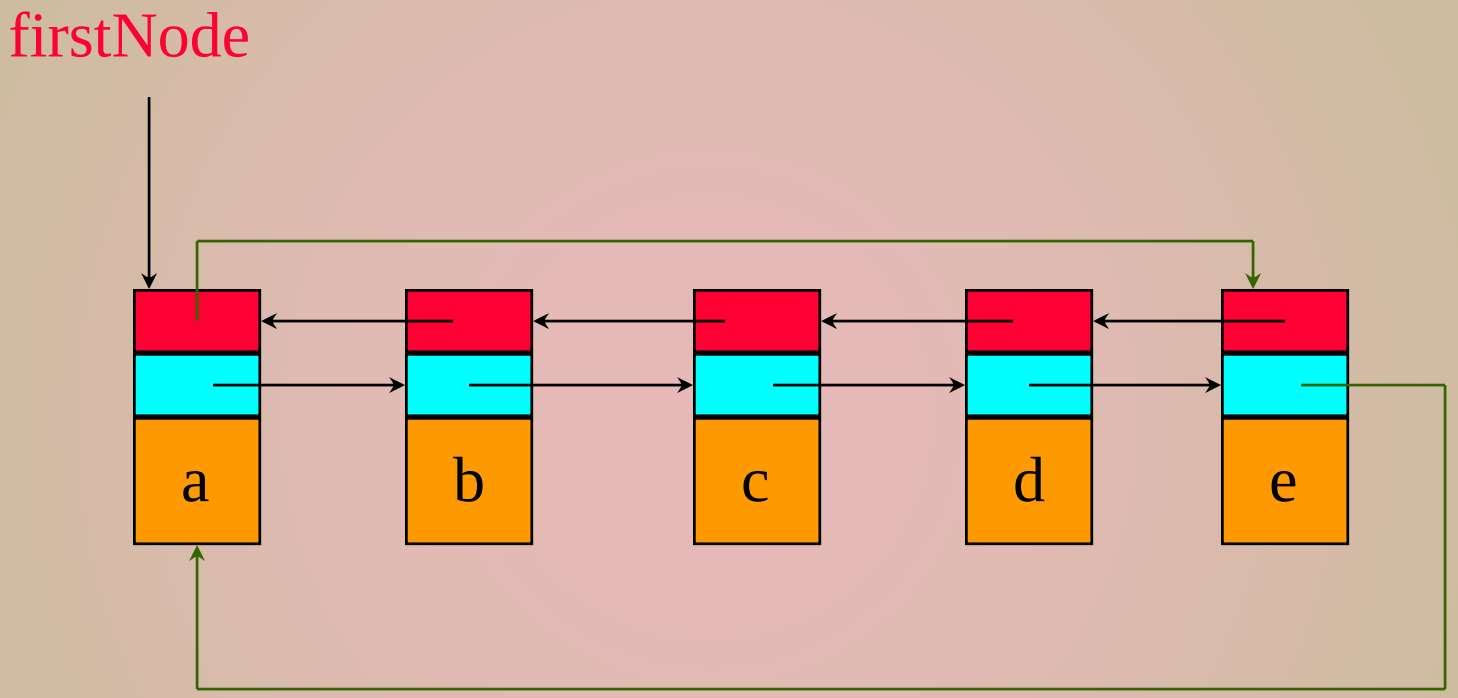



Circular List
firstNode
a
b
c
d
e
headerNode
NULL

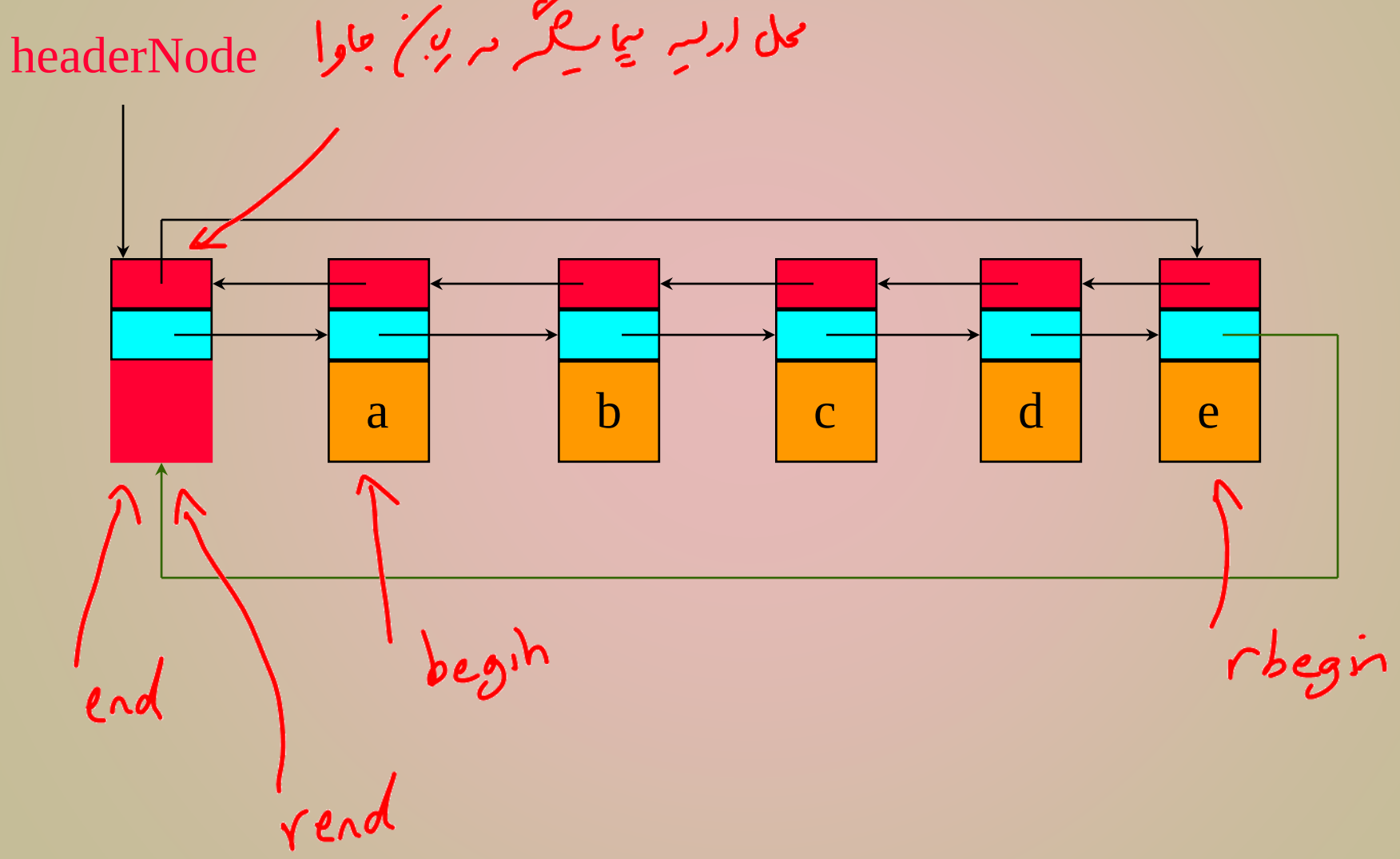
Doubly Linked List



Doubly Linked Circular List



Doubly Linked Circular List With Header Node



Empty Doubly Linked Circular List With Header Node