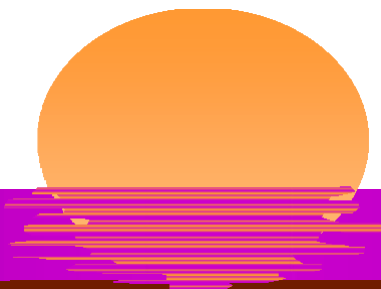


# Ү.СS203 - Өгөгдлийн бүтэц ба алгоритм 2022-2023

Д.Золбоо  
МТ-н салбарын багш





- Шугаман жагсаалт.
- Нэг төгсгөл нь **front** - нүүр.
- Нөгөө төгсгөл нь **rear** - сүүл.
- Нэмэхдээ зөвхөн **сүүлд** нь.
- Устгахдаа зөвхөн **нүүрнийх** нь.

# Автобусны буудлын дараалал



# Автобусны буудлын дараалал



front



rear



# Автобусны буудлын дараалал



front



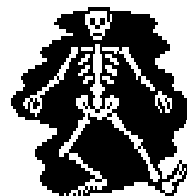
rear



# Автобусны буудлын дараалал



front



rear



# Интерфейс - Queue

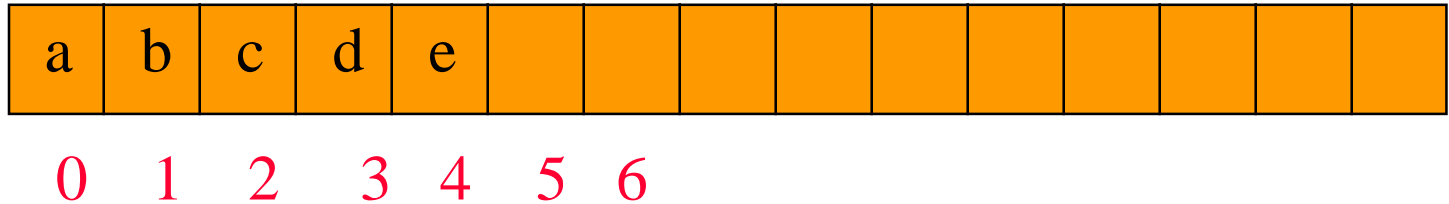
```
public interface Queue
{
    public boolean isEmpty();
    public Object getFrontEelement();
    public Object getRearEelement();
    public void put(Object theObject);
    public Object remove();
}
```

# Стек хэрэглэсэн жишээг эргэж харья

- Стекыг дарааллаар сольж болохгүй жишээ.
  - Хаалт хослох.
  - Ханойн цамхаг.
  - Switchbox routing.
  - Аргыг дуудах ба буцах.
  - Try-catch-throw хэрэгжүүлэлт.
- Стекыг дарааллаар сольж болох жишээ.
  - Төөрөлдсөн оготно.
    - Үр дүн нь гарах дөт замыг олох.



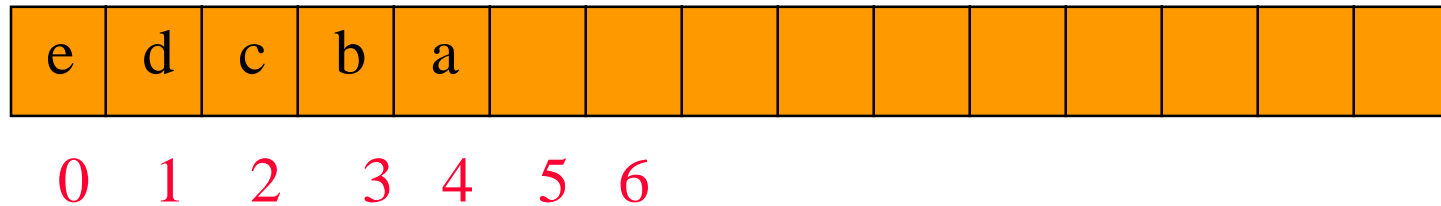
# ArrayList –ээс удамших



➤ Жагсаалтын зүүн төгсгөл нүүр, баруун төгсгөл нь сүүл бол

- `Queue.isEmpty() => super.isEmpty()`
- `getFrontElement() => get(0)`
- `getRearElement() => get(size() - 1)`
- `put(theObject) ==> add(size(), theObject)`
- `remove() ==> remove(0)`

# ArrayLinearList –ээс удамших



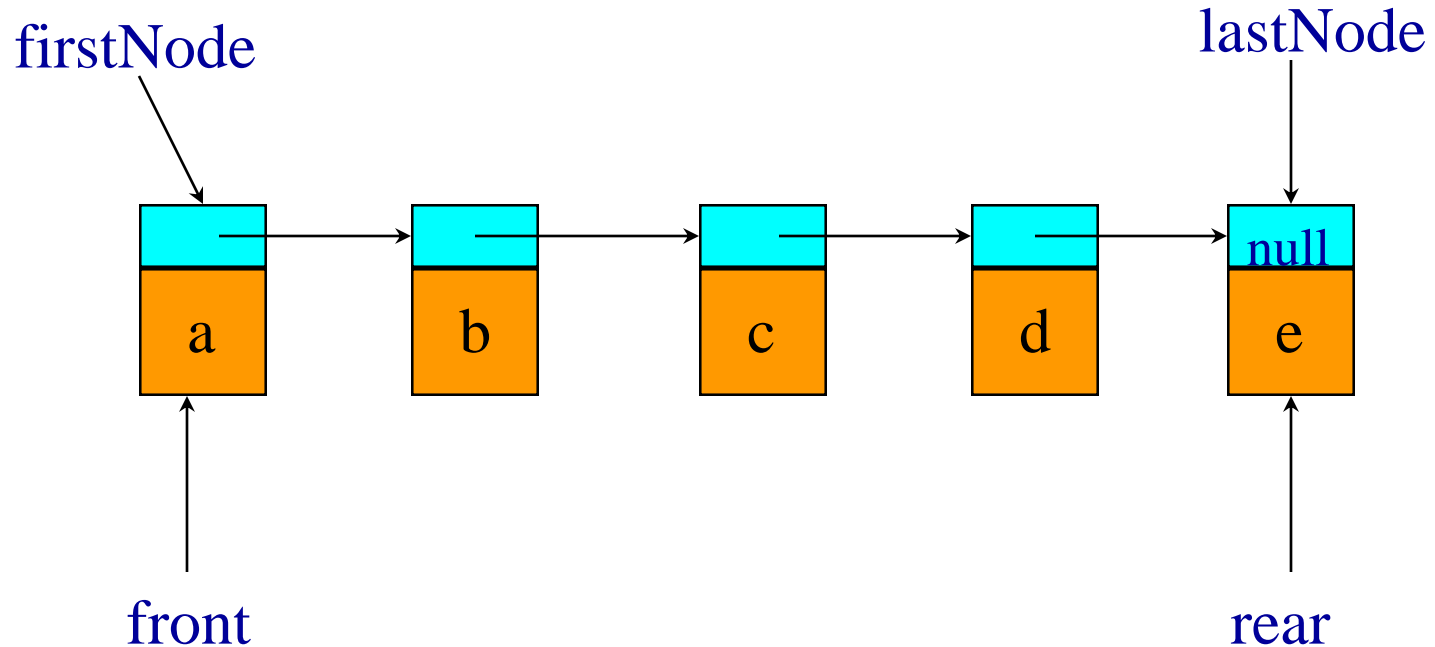
— Жагсаалтын зүүн төгсгөл сүүл, баруун төгсгөл нь нүүр бол

- `Queue.isEmpty() => super.isEmpty()`
- `getFrontElement() => get(size() - 1)`
- `getRearElement() => get(0)`
- `put(theObject) => add(0, theObject)`
- `remove() => remove(size() - 1)`

# ArrayList –ээс удамших

- Үйлдлүүдийг гүйцэтгэх хугацаа богино (массивыг хоёр дахин ихэсгэхээс бусад), бидэнд сайжруулсан массив дүрслэл хэрэгтэй

# ExtendedChain –ээс удамших

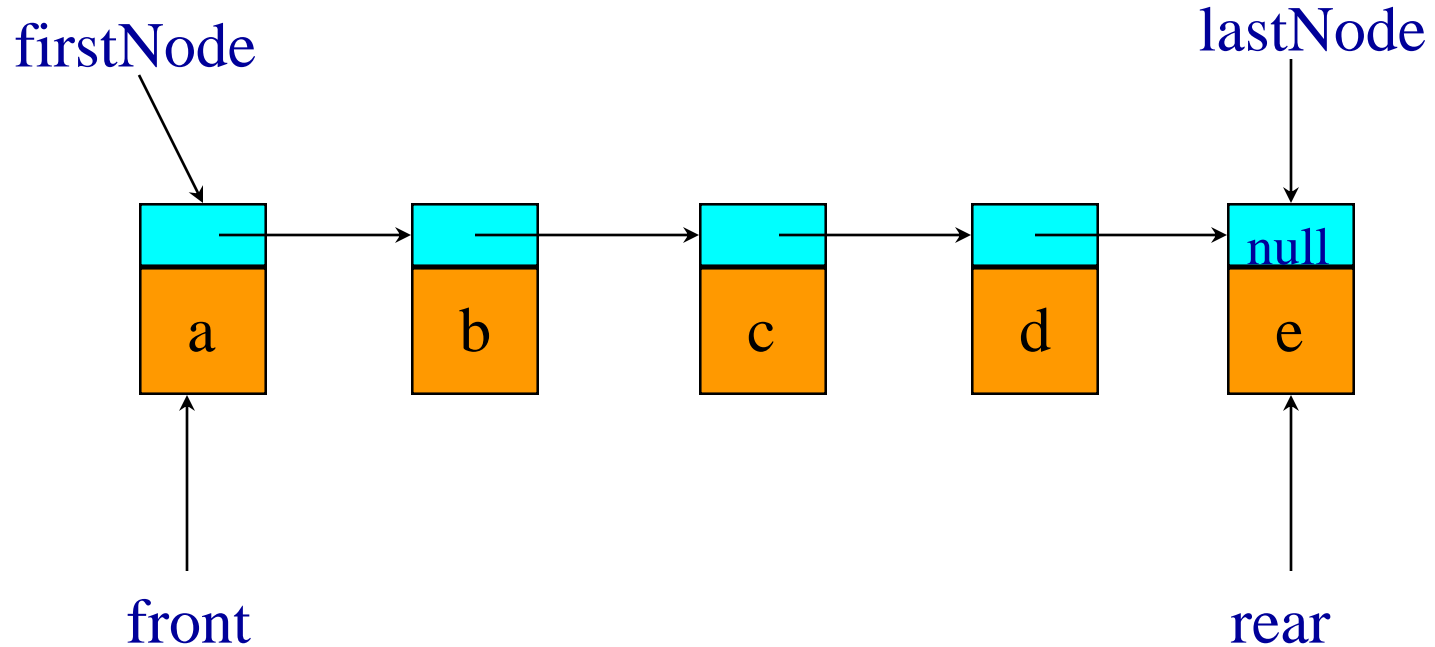


➤ жагсаалтын зүүн төгсгөл нь нүүр, баруун төгсгөл нь сүүл бол

- `Queue.isEmpty() => super.isEmpty()`

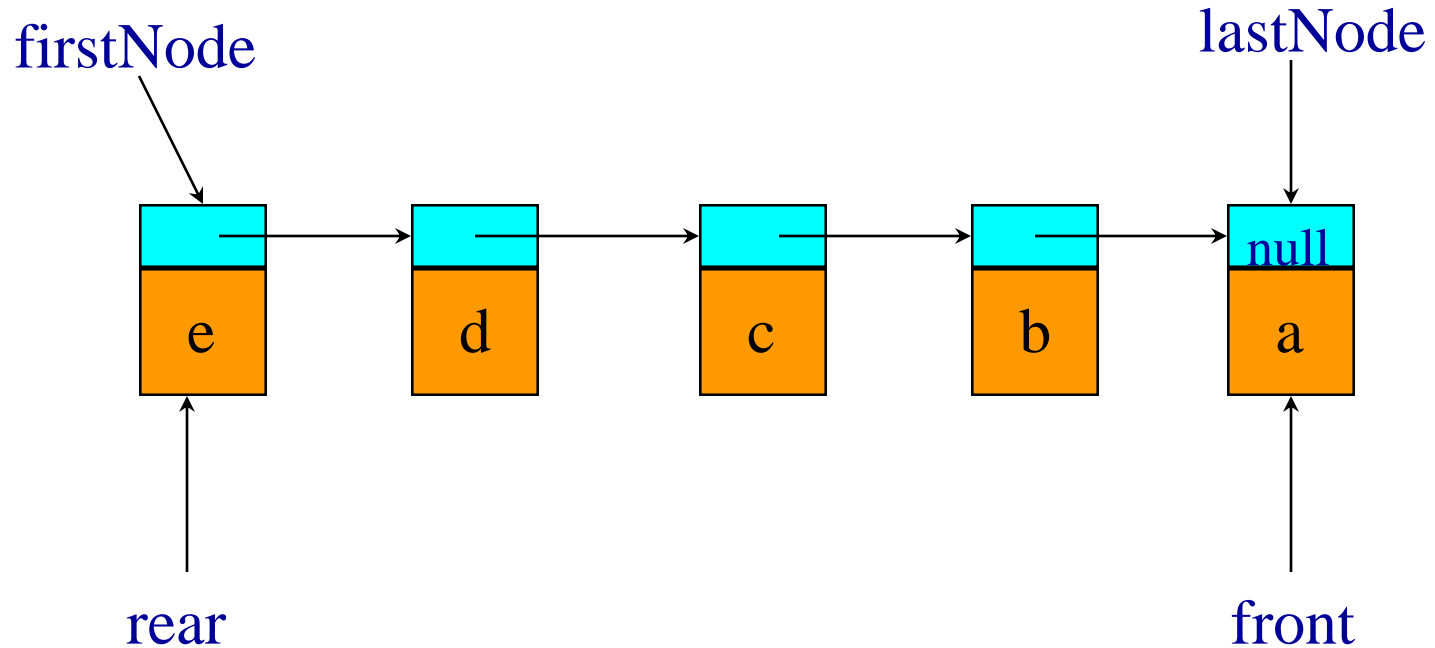
- `getFrontElement() => get(0)`

# ExtendedChain –ээс удамших



- `getRearElement()` => `getLast()` ... шинэ арга
  - `put(theObject)` => `append(theObject)`
  - `remove()` => `remove(0)`

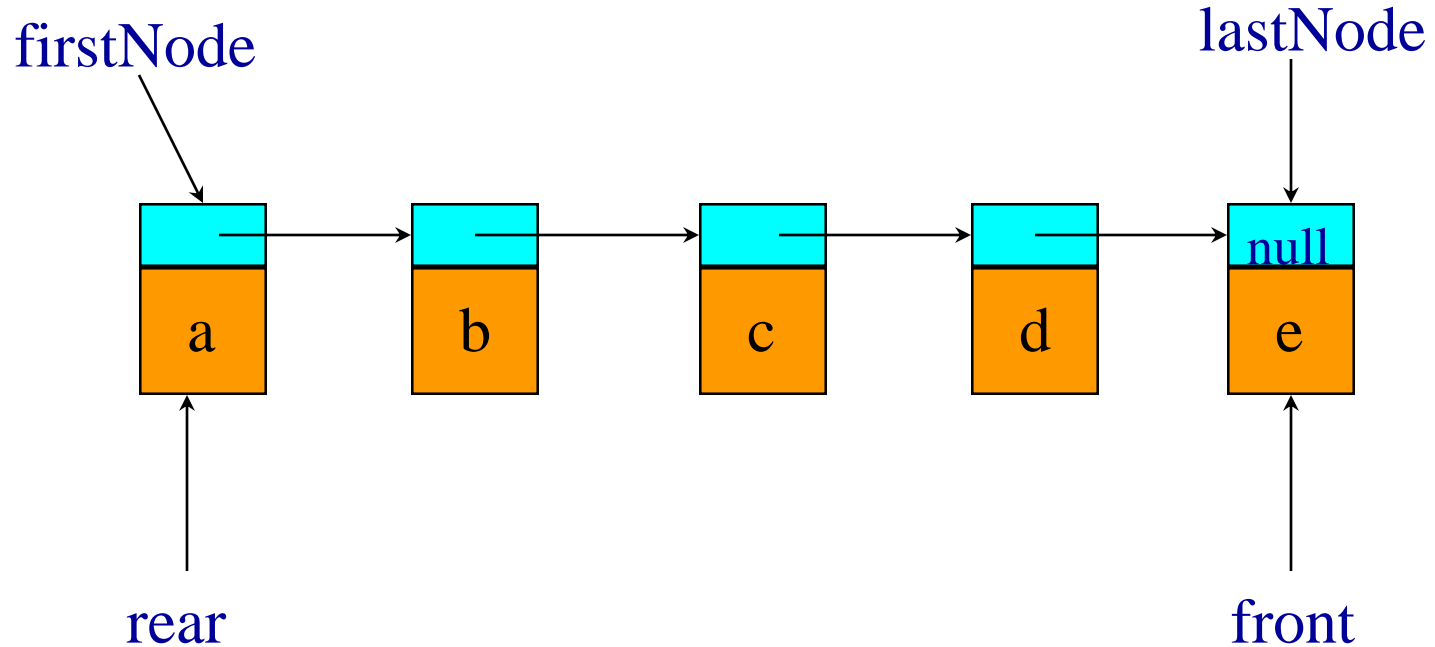
# Derive From ExtendedChain



➤ Жагсаалтын зүүн төгсгөл сүүл, баруун төгсгөл нь нүүр бол

- `Queue.isEmpty() => super.isEmpty()`  
– `getFrontElement() => getLast()`

# ExtendedChain –ээс удамших



- `getRearElement()` => `get(0)`
  - `put(theObject)` => `add(0, theObject)`
  - `remove()` => `remove(size-1)`

# Өөрчилсөн холбоост код

- **ExtendedChain** –ээс уламжилснаас сайн үзүүлэлт хэрэгтэй бол **Queue** –д зориулсан холбоост классыг эхнээс нь кодчилох хэрэгтэй

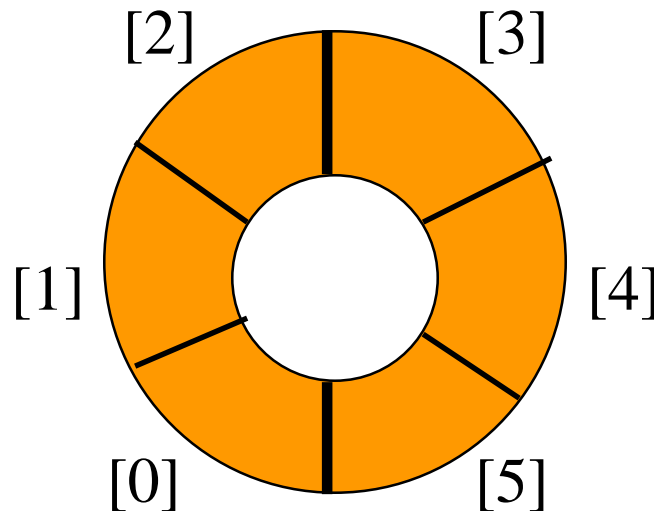


# Өөрчилсөн массив дараалал

- 1D массив **queue** -г ашиглая

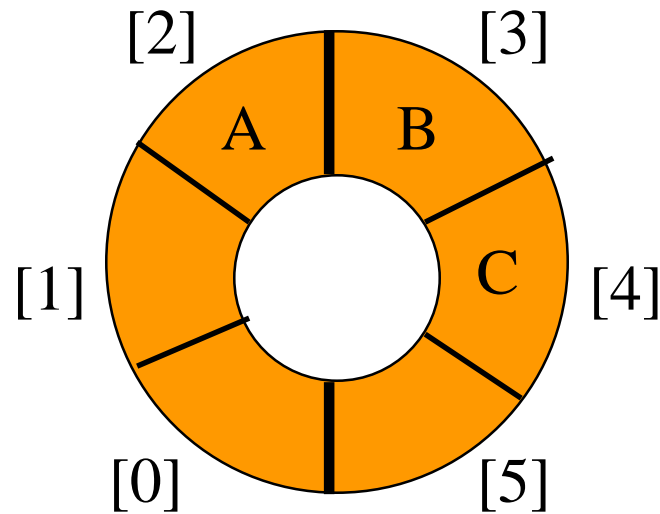
**queue[]** 

- Массивын цагираг харагдац.



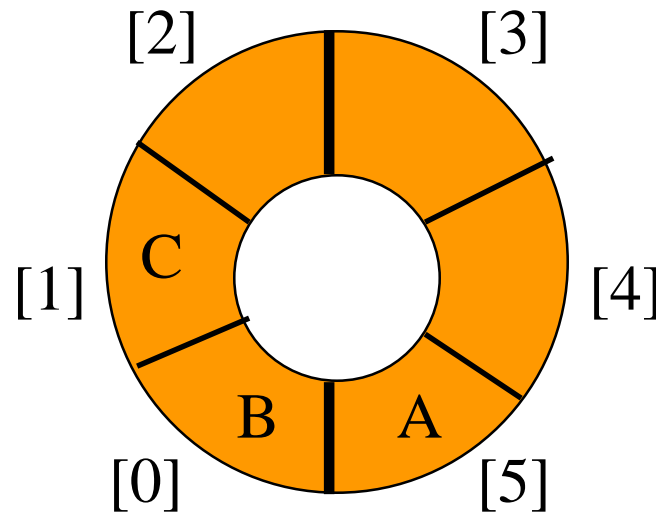
# Өөрчилсөн массив дараалал

- 3 элементтэй хувилбар.



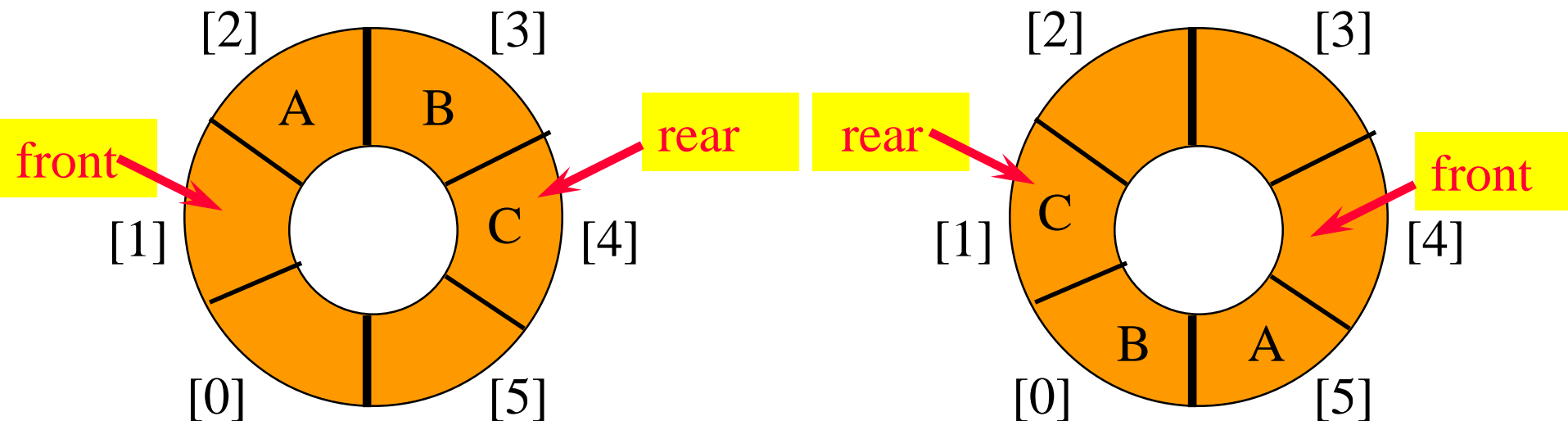
# Өөрчилсөн массив дараалал

- 3 элементтэй өөр нэг хувилбар.



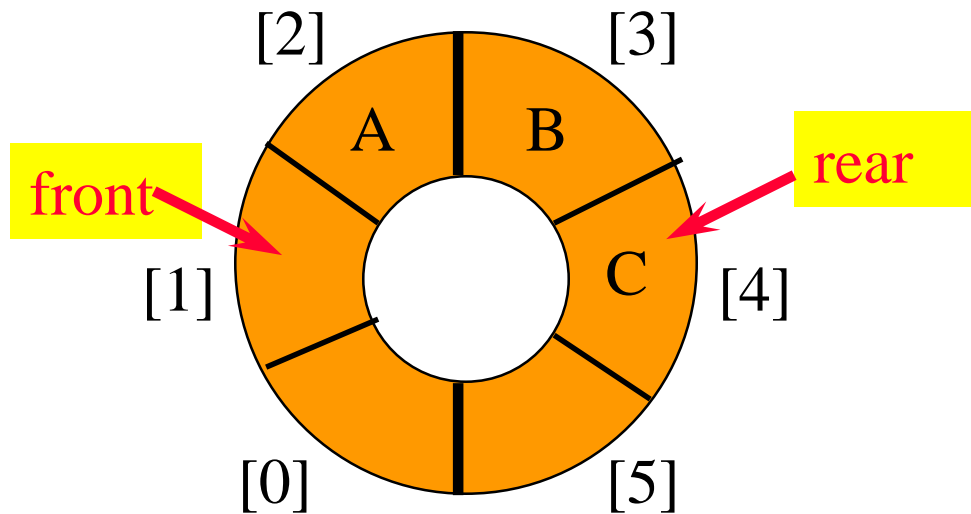
# Өөрчилсөн массив дараалал

- Бүхэл хувьсагч **front** , **rear** -г ашиглая
  - **front** эхний элементээс цагийн зүүний дагуу явсан нэг байршил
  - **rear** сүүлийн элементийн байршлыг өгнө



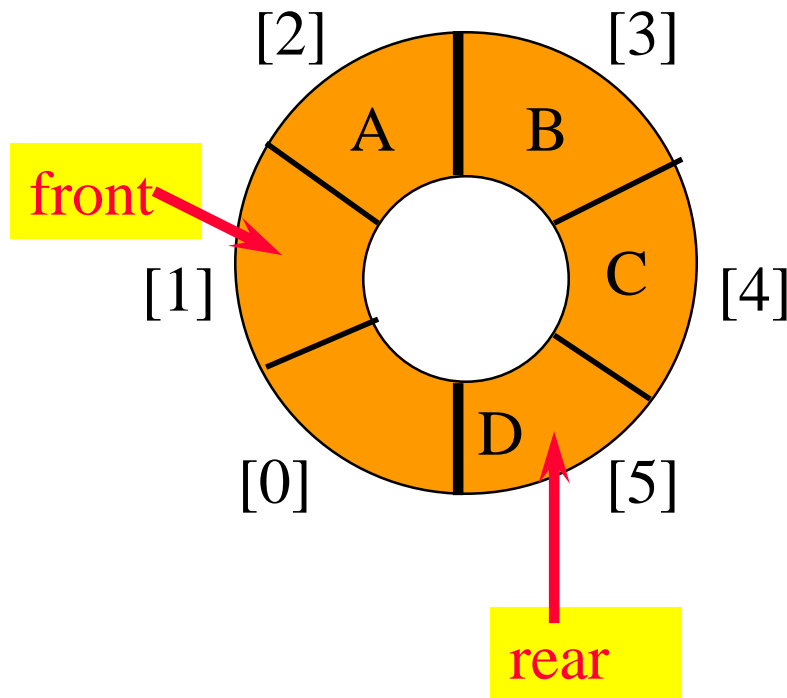
# Элемент нэмэх

- **rear** цагийн дагуу нэгээр хөдөлгөнө.



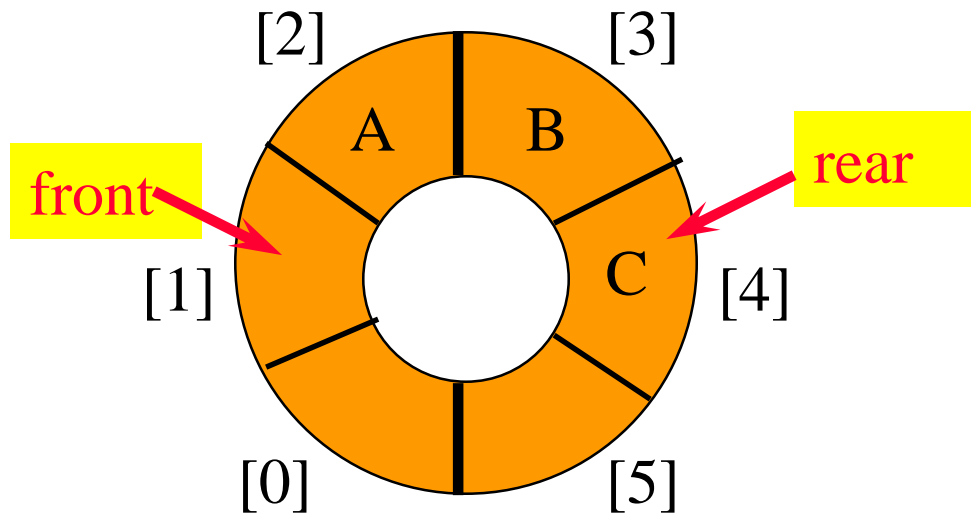
# Элемент нэмэх

- **rear** цагийн дагуу нэгээр хөдөлгөнө.
- **queue[rear]** -д элементийг хийнэ



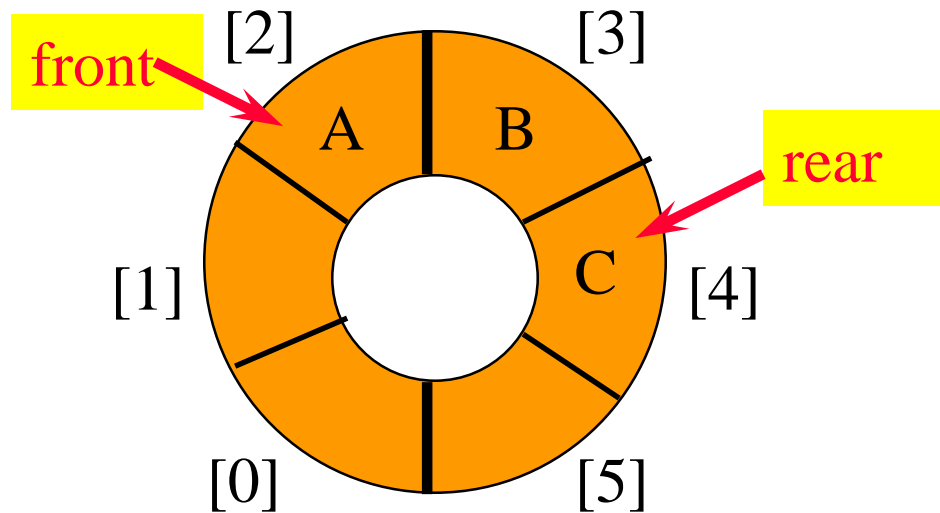
# Элементийг устгах

- **front** цагийн дагуу нэгээр хөдөлгөнө.



# Элемент устгах

- **front** цагийн дагуу нэгээр хөдөлгөнө.
- **queue[front]** -аас элемент авна.

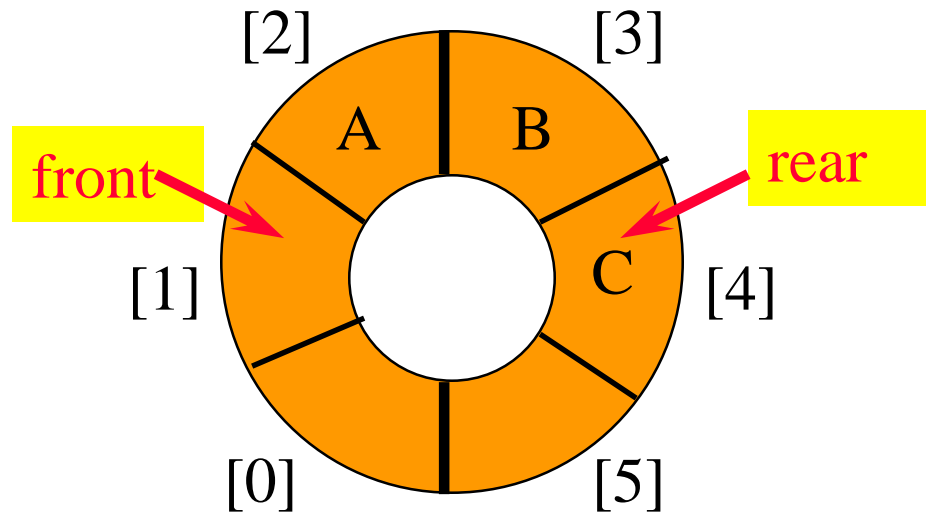




# rear –г цагийн дагуу хөдөлгөх

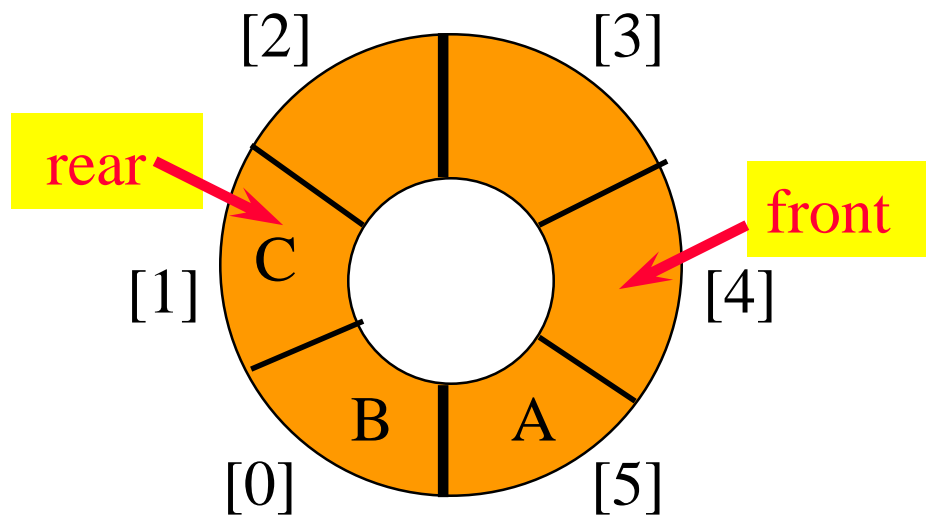
- `rear++;`

`if (rear == queue.length) rear = 0;`

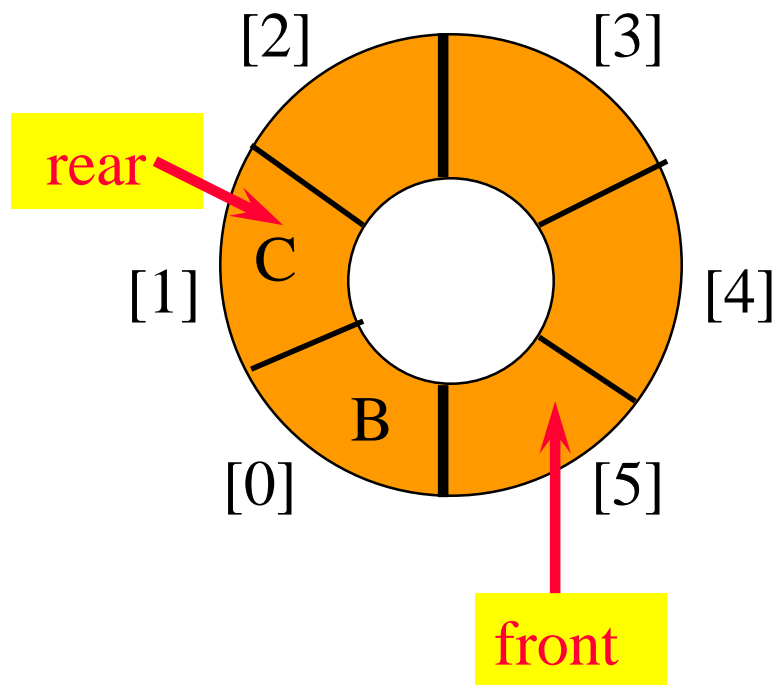


- `rear = (rear + 1) % queue.length;`

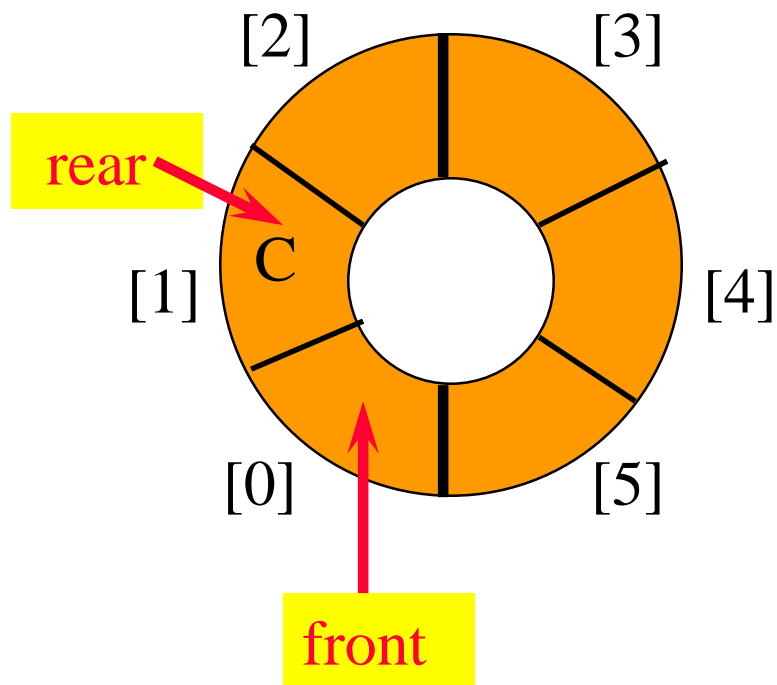
# Дарааллыг хоослох



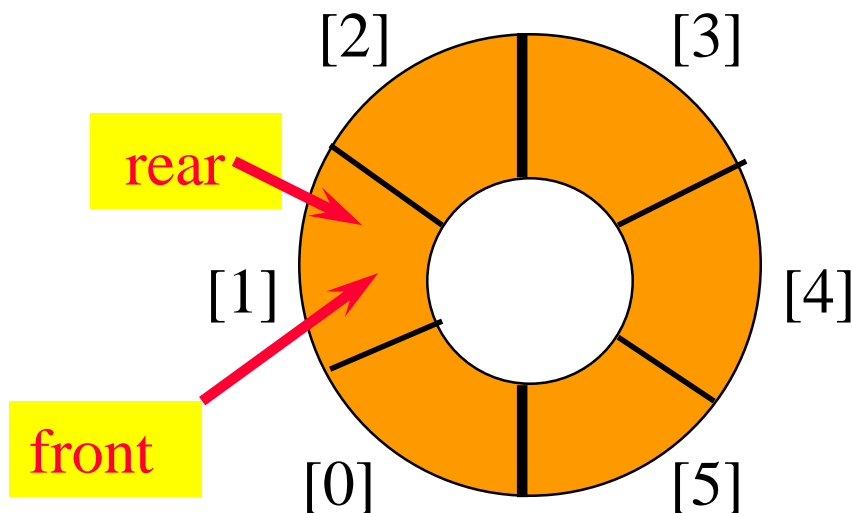
# Дарааллыг хоослох



# Дарааллыг хоослох

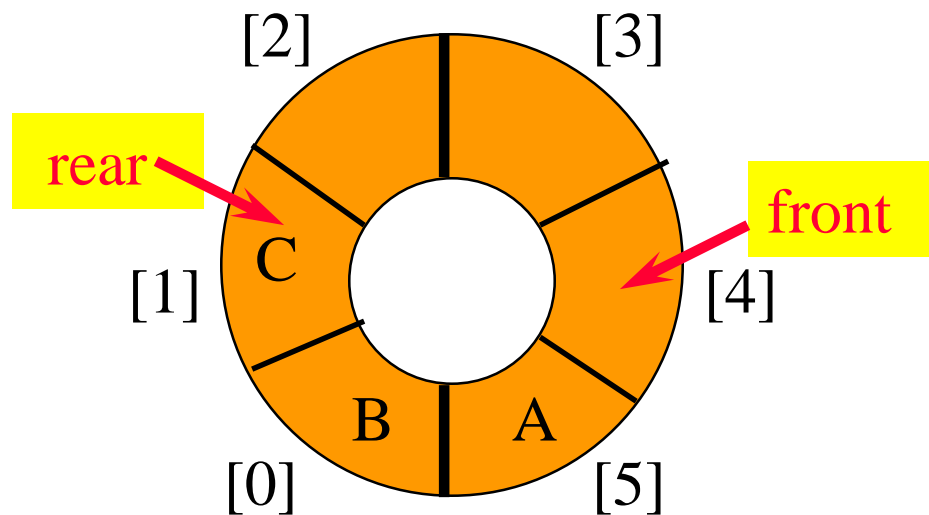


# Дарааллыг хоослох

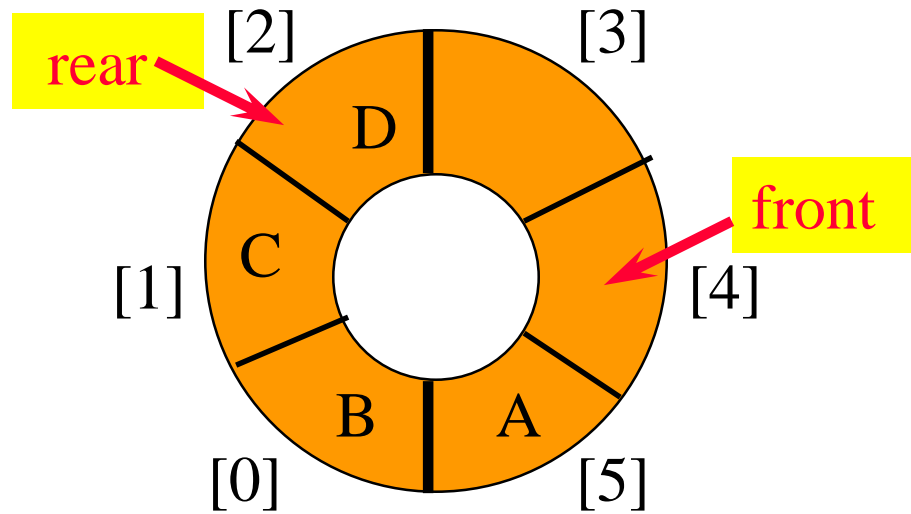


- Дараалсан устгалтуудаар  $front = rear$  болоход дараалал хоослогдоно
- Анх байгуулагдахад хоосон байна.
- Иймд эхэндээ  $front = rear = 0$ .

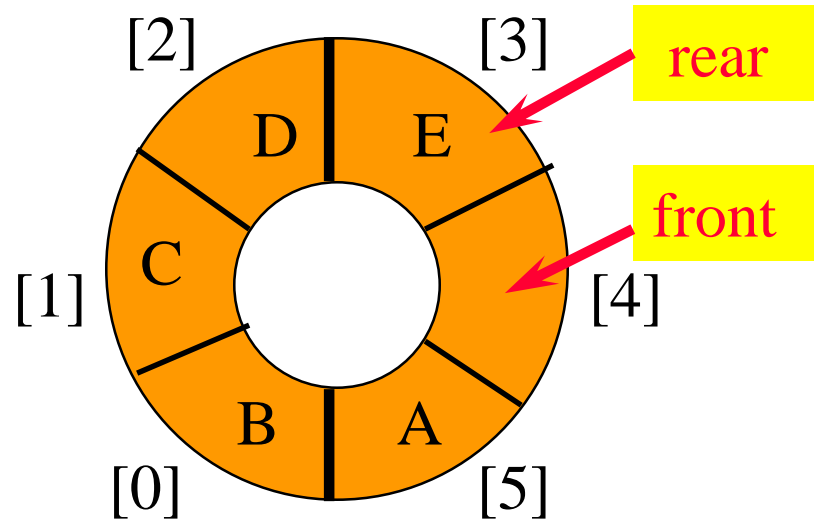
# Дарааллын савыг дүүргэх



# Дарааллын савыг дүүргэх

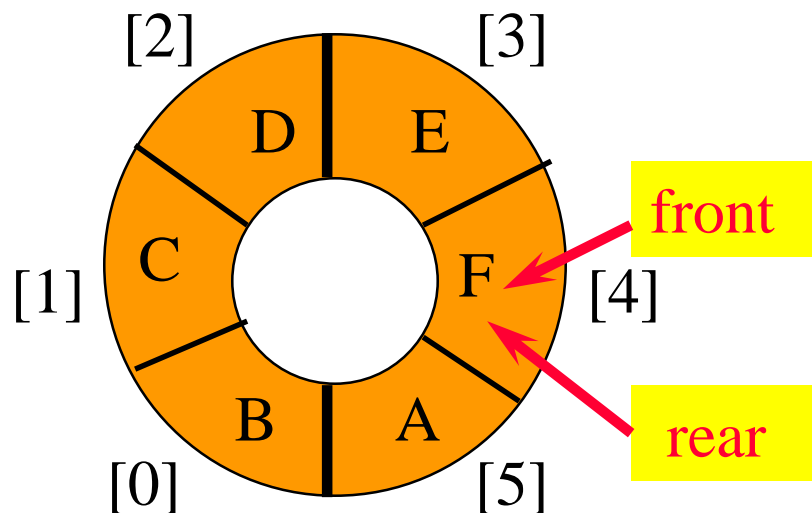


# Дарааллын савыг дүүргэх





# Дарааллын савыг дүүргэх



- Дараалсан нэмэлтүүдээр  $front = rear$  болоход сав дүүрнэ
- Сав дүүрэн үү, хоосон уу гэдгийг ялгах хэцүү боллоо!

# Анхаар!!!!

- Засвар.

- Дарааллыг бүү дүүргэ.

- Нэмэгдэх элемент дарааллыг дүүргэх бол массивын хэмжээг нэмэгдүүл.
    - Үүнийг сурах бичигт үзүүлсэн.

- Булын хувьсагч **lastOperationIsPut** -г ашигла

- **put** үйлдэл бүрийн дараа **true** болго
    - **remove** үйлдэл бүрийн дараа **false** болго
    - дараалал хоосон -> **(front == rear) && !lastOperationIsPut**
    - Дараалал дүүрэн -> **(front == rear) && lastOperationIsPut**

# Анхаар!!!!

- Засвар (үргэлжлэл).
  - Бүхэл хувьсагч **size** -г ашигла
    - **put** үйлдэл бүрийн дараа **size++**.
    - **remove** үйлдэл бүрийн дараа **size--**.
    - Дараалал хоосон -> (**size == 0**)
    - Дараалал дүүрэн -> (**size == queue.length**)
  - Эхний хувилбараар чанарын үзүүлэлт арай дээр байх болно.