

فصل چہارم

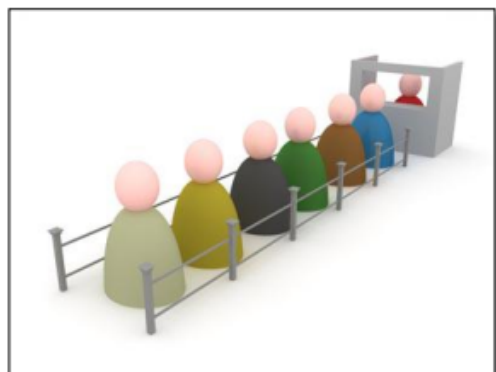
صف

نجمہ منصوری

صف

□ تفاوت **صف** با پشته در این است که درج و حذف عناصر در صف از اصل «اولین ورودی، اولین خروجی» پیروی می‌کند.

□ به عبارت دیگر هر زمان که بخواهیم می‌توانیم یک عنصر به انتهای صف اضافه کنیم، اما در هنگام حذف تنها اجازه داریم اولین عنصر (عنصری که بیشتر از بقیه‌ی عناصر در صف بوده است) را از صف حذف کنیم.

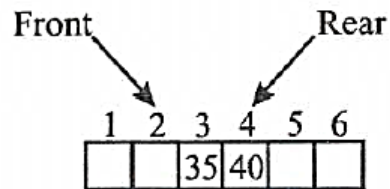


□ عملیات اصلی صف.

□ درج یک عنصر در **انتهای** صف

□ حذف یک عنصر از **ابتدای** صف

در صورتی که از آرایه n تایی $Q[1..n]$ برای نمایش و ذخیره ساختار یک صف استفاده شود همیشه از دو اشاره گر Front و Rear که به ترتیب به ابتدا و انتهای صف اشاره می کنند استفاده می کنیم.

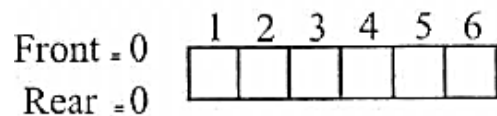


باتوجه به شکل دیده می شود که :

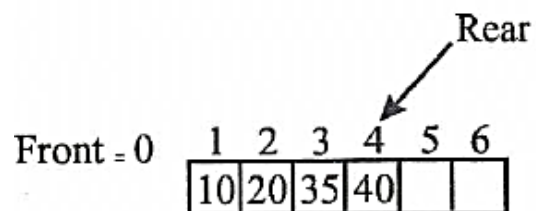
Front : به خانه قبل از خانه اول صف اشاره می کند.
Rear : به خانه آخر صف اشاره می کند.

مثال :

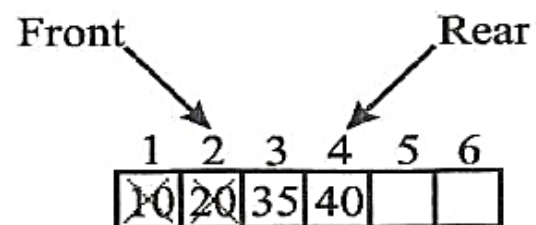
(۱) صف 6 عضوی روبرو را در نظر بگیرید.



(۲) درج عناصر 10 , 20 , 35 , 40



(۳) حذف 2 داده (به طور عادی داده‌ها از ابتدا حذف می‌شوند).



در وضعیت بالا با این که 2 داده از ابتدای صف حذف شده‌اند اما فقط دو خانه 5 , 6 برای درج وجود دارند چون حرکت rear به سمت جلو بوده و امکان استفاده از خانه‌های قبلی را ندارد.

شرایط مرزی و اولیه: (مهم)

Front: همیشه به خانه قبل از خانه اول صف اشاره کند.

Rear: همیشه به خانه آخر صف اشاره کند.

در صورتی که

انتخاب آرایه $Q[1..n]$ برای پیاده‌سازی صف خطی دارای شرایط اولیه و مرزی زیر است:

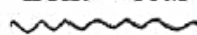
پر بودن صف خطی

$$\text{rear} = n$$



خالی بودن صف خطی

$$\text{front} = \text{rear}$$



وضعیت اولیه صف

$$\text{front} = \text{rear} = 0$$

عملیات در صف (Queue)

۱- درج در صف

```
procedure insert (var item : data type);  
begin  
  if rear = n then  
    write ('Queue is Full')  
  else  
    begin  
      rear := rear + 1;  
      Q [rear] := item;  
    end;  
end;
```

دیده می شود که چون rear به خانه آخر صف اشاره می کند برای عمل درج ابتدا rear یک واحد به جلو حرکت کرده سپس داده item در خانه خالی انتهای صف درج می شود.

۲- حذف از صف

```
procedure delete (var item:data type);  
begin  
  if front = rear then  
    write ('Queue is Empty')  
  else  
    begin  
      front := front + 1;  
      item := Q [front];  
    end;  
  end;  
end;
```

دیده می شود که چون front همیشه به قبل از اول صف اشاره می کند برای عمل حذف ابتدا front 1 واحد به جلو حرکت کرده سپس داده ابتدای صف را خارج کرده در item قرار می دهد.

مشکل صف‌های خطی

حرکت تدریجی عناصر به سمت انتهای صف و عدم امکان استفاده از خانه‌های ابتدای صف که در اثر حذف خالی شده‌اند.

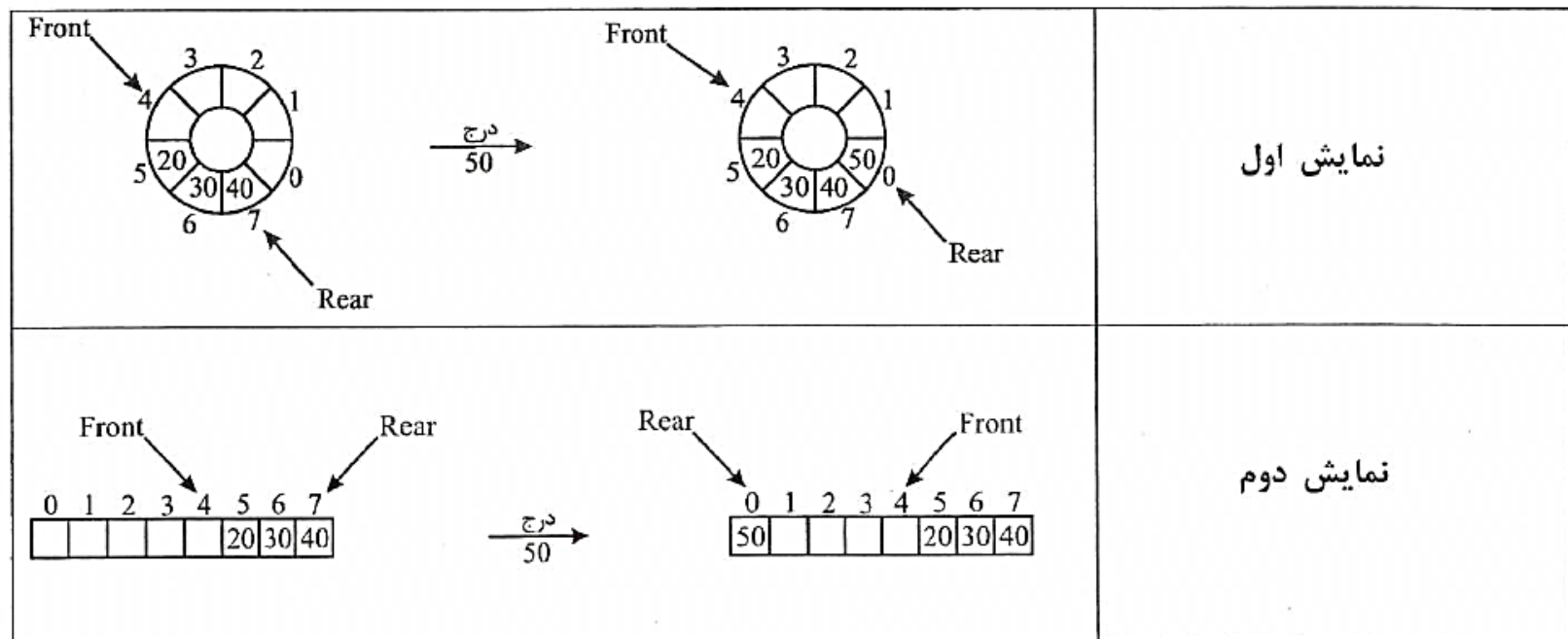
راه حل رفع مشکل صف‌های خطی

۱- **شیفت خانه‌های انتهای صف** به سمت ابتدای صف که این عمل در صورتی که تعداد خانه‌های صف زیاد باشد هزینه بالایی خواهد داشت.

۲- استفاده از **صف حلقوی** به این مفهوم که زمانی که به انتهای صف رسیدیم عمل درج را دوباره از ابتدای صف انجام می‌دهیم و این حرکت چرخشی می‌باشد.
روش دوم در عمل مقرون به صرفه‌تر بوده و استفاده می‌شود.

صف حلقوی (Circular Queue)

آرایه n تایی $Q[0..n-1]$ را می‌توان به صورت یک صف حلقوی در نظر گرفت به طوری که در این صف زمانی که $rear$ برابر $n-1$ می‌شود، عنصر بعدی در خانه 0 قرار می‌گیرد.



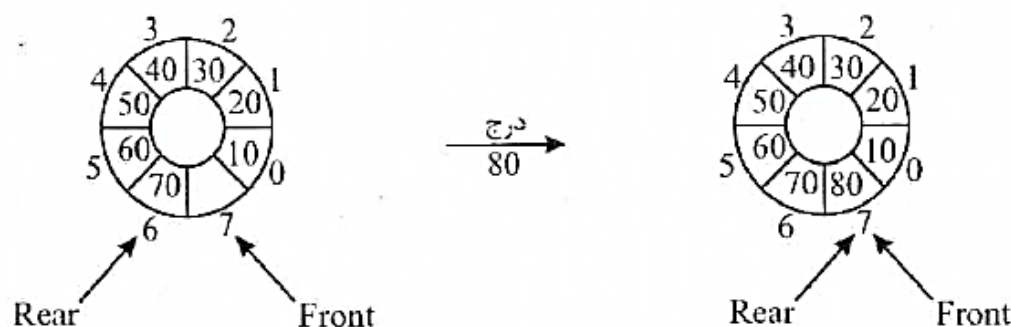
شرایط مرزی در صف حلقوی

پر (Full)	خالی (Empty)
$\text{Front} = (\text{Rear} + 1) \bmod n$	$\text{Front} = \text{Rear}$

تعداد خانه‌های قابل استفاده در صف حلقوی

در هر صف حلقوی $Q[0 .. n - 1]$ با ظرفیت n ، همیشه یک خانه باید خالی باشد، و حداکثر از $n - 1$ خانه صف می‌توان استفاده کرد، این به آن علت است که بتوان وضعیت پر یا خالی بودن صف حلقوی را تشخیص داد.

دقت کنید: در صورتی که در یک صف حلقوی $Q[0 .. n - 1]$ با ظرفیت n از همه خانه‌های صف استفاده کنیم و یک خانه را خالی نگذاریم وضعیت پر یا خالی بودن صف حلقوی قابل تشخیص نخواهد بود.



در مثال قبل بعد از درج 80 در صف حلقوی و استفاده از همان یک خانه‌ای که باید در صف خالی می‌ماند $\text{Front} = \text{Rear} = 7$ می‌شود که همان شرط خالی بودن صف است در صورتی که صف پر شده و این یک تناقض است برای نبودن تناقض فوق و امکان تشخیص وضعیت پر یا خالی بودن صف حلقوی یک خانه را خالی نگه می‌داریم.

عملیات در صف حلقوی

۱- حذف از صف حلقوی

```
procedure delqueue (item: type);  
begin  
if front = rear then  
write('queue is Empty')  
else  
begin  
front := (front + 1) mod n;  
item := Q[front];  
end;  
end;
```

۲- درج در صف حلقوی

```
procedure Addqueue (item:type);  
begin  
if  $(rear + 1) \bmod n = front$  then  
write('queue is Full')  
else  
begin  
 $rear := (rear + 1) \bmod n;$   
Q[rear]:=item;  
end;  
end;
```

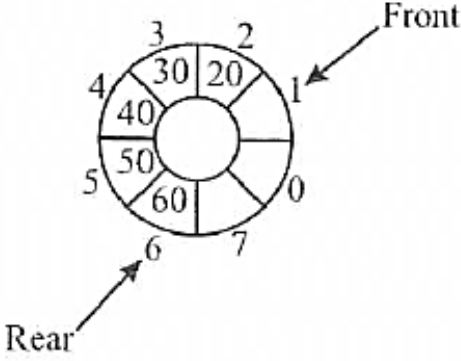
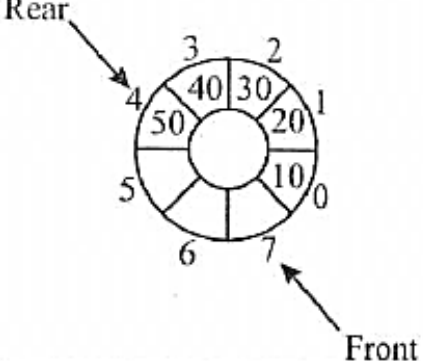
در هر دو روش حذف و درج به ترتیب از جملات $rear := (rear + 1) \bmod n$ و $front := (front + 1) \bmod n$ استفاده شده است که علت استفاده از mod در شرایطی است که front یا rear به خانه آخر اشاره کرده باشند و حرکت به جلو آن‌ها را به ابتدای صف منتقل می‌کند.

تعداد خانه‌های پر و خالی یک صف حلقوی با ظرفیت n

الف) $Front$ به قبل از اول صف اشاره می‌کند:

ردیف	وضعیت	تعداد خانه های صف	تعداد خانه های خالی
۱	$Rear \geq Front$	$Rear - Front$	$n - (Rear - Front)$
۲	$Rear < Front$	$n - (Front - Rear)$	$Front - Rear$

-مثال : برای یک صف حلقوی زیر با ظرفیت $n = 8$ همواره داریم:

	<p>Rear \geq Front</p>	<p>خانه های صف: $\text{Rear} - \text{Front} = 5$ خانه های خالی: $n - (\text{Rear} - \text{Front}) = 3$</p>
	<p>Front $>$ Rear</p>	<p>خانه های صف: $n - (\text{Front} - \text{Rear}) = 5$ خانه های خالی: $\text{Front} - \text{Rear} = 3$</p>

ب) Front به اول صف اشاره می‌کند:

ردیف	وضعیت	تعداد خانه های صف	تعداد خانه های خالی
۱	$Rear \geq Front$	$Rear - Front + 1$	$n - (Rear - Front + 1)$
۲	$Rear < Front$	$n - (Front - Rear - 1)$	$Front - Rear - 1$

عناصر صف‌های Q_1 و Q_2 از چپ به راست به صورت زیر است (عنصر سمت چپ ابتدای صف است):

$$Q_1 = 10, 25, 17, 41, 19, 26, 75$$

$$Q_2 = 1, 5, 7, 4, 9, 6$$

$\text{makenull}(Q_3)$

اگر x و y عناصر صف باشند، پس از اجرای قطعه برنامه زیر:

$i=0$

while (not empty(Q_1) and not empty(Q_2))

$i = i + 1$

$x = \text{DeleteQ}(Q_1)$

$y = \text{DeleteQ}(Q_2)$

if ($y = i$) **then** AddQ(Q_3, x)

endwhile

محتوی صف Q_3 برابر است با:

$$Q_3 = 1, 5, 7 \text{ (۴)}$$

$$Q_3 = 10, 41, 26 \text{ (۳)}$$

$$Q_3 = 10, 25, 17 \text{ (۲)}$$

$$Q_3 = 1, 4, 6 \text{ (۱)}$$

گزینه «۳» در این قطعه برنامه ابتدا یک صف خالی به نام Q_3 ایجاد می‌شود. سپس در حلقه `while` تا زمانی که صف Q_1 و صف Q_2 هیچ‌کدام خالی نباشد یکی به شمارنده i اضافه شده و دو عنصر یکی از صف Q_1 و دیگری از صف Q_2 حذف می‌شود و به ترتیب در x و y قرار می‌گیرد. اگر مقدار y با شمارنده i برابر باشد، مقدار x در صف Q_3 قرار خواهد گرفت. در این صورت خواهیم داشت:

i	x	y	شرط حلقه	شرط $y = i$	Q_3
0	–	–	✓		–
1	10	1	✓	✓	10
2	25	5	✓	×	10
3	17	7	✓	×	10
4	41	4	✓	✓	10, 41
5	19	9	✓	×	10, 41
6	26	6	×	✓	10, 41, 26

تابع زیر داده ITEM را به صف اضافه می کند. چنانچه صف را توسط یک آرایه چرخشی پیاده سازی نماییم در جای خالی چه دستوری باید

نوشته شود؟

(مهندسی کامپیوتر - آزاد ۸۹)

QINSERT(Queue, N, FRONT, REAR, ITEM)

If FRONT = 1 and REAR = N, or if FRONT = REAR + 1, then :

write : OVERFLOW, and Return.

If FRONT = NULL, then :

Set FRONT := 1 and REAR := 1

ELSE if REAR = N then :

Set

Else :

Set REAR := REAR + 1

Set QUEUE[REAR] := ITEM.

Return.

REAR = FRONT = 2 (۴)

REAR := N + 1 (۳)

REAR := 1 (۲)

REAR := N - 1 (۱)

گزینه «۲» در این برنامه اگر $rear = n$ و $front = 1$ باشد یا $front = rear + 1$ پیغام overflow صادر می شود و اگر $front = Null$ ، یعنی صف خالی باشد $front = rear = 1$ می شود و اگر $front \neq Null$ و $Rear = N$ باشد، چون شرط اول نیز برقرار نبوده که به این شرط رسیده، پس $front \neq 1$ است و می توانیم از عنصر اول برای درج استفاده کنیم پس $Rear = 1$ صحیح است.

بعضی از کاربردهای صف

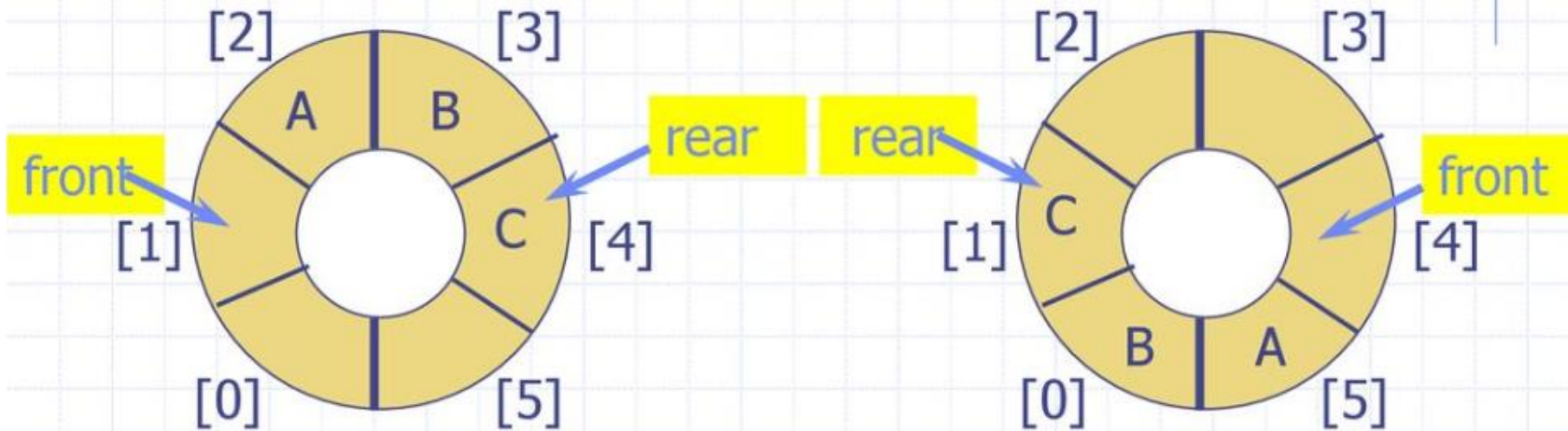
۱- در صف پردازش‌های سیستم عامل

۲- صف چاپگر

۳- پیمایش **BFS = Breath First Search** (سطحی، ردیفی، پهنا) در درخت و گراف

صف حلقوي

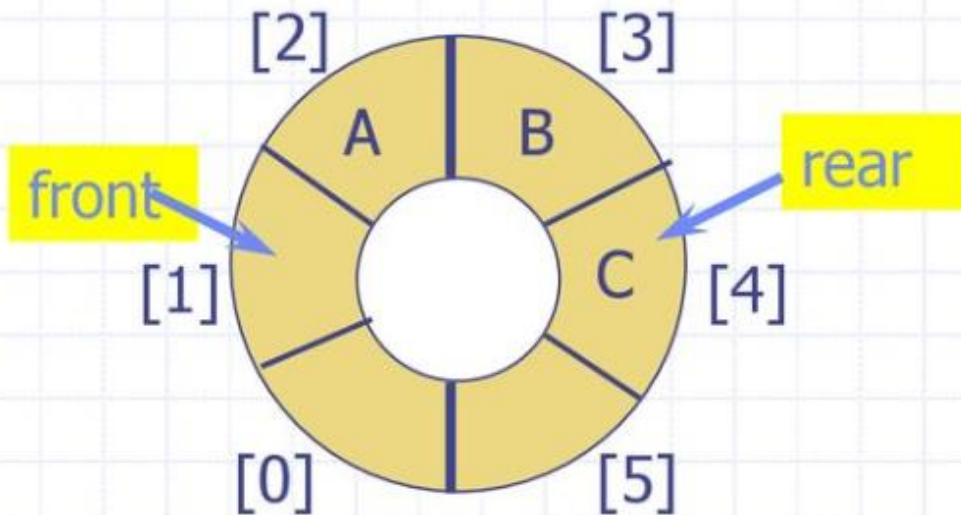
front
rear
یک محل قبل از اولین عنصر
محل آخرین عنصر



صف حلقوي

■ اضافه کردن به صف

rear را در جهت عقربه های ساعت یک واحد افزایش دهید

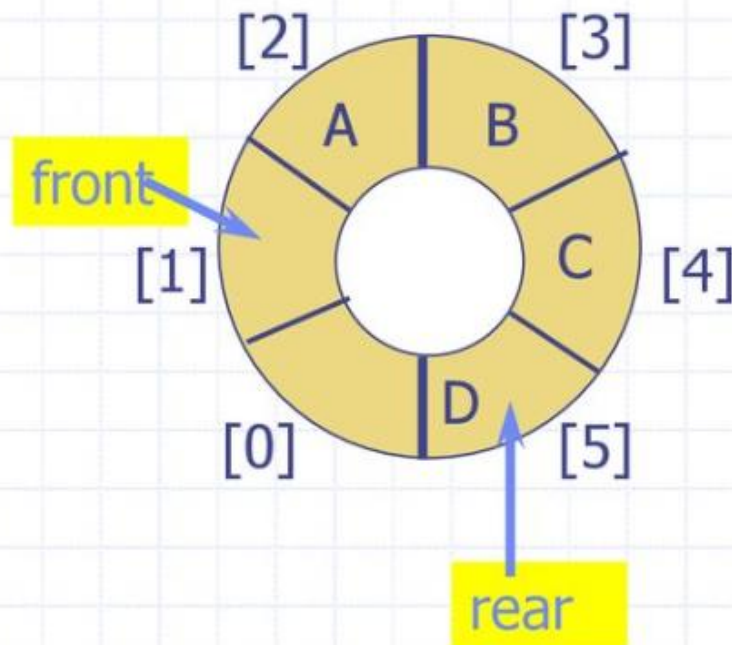


صف حلقوي

■ اضافه کردن به صف

rear را در جهت عقربه های ساعت یک واحد افزایش دهید

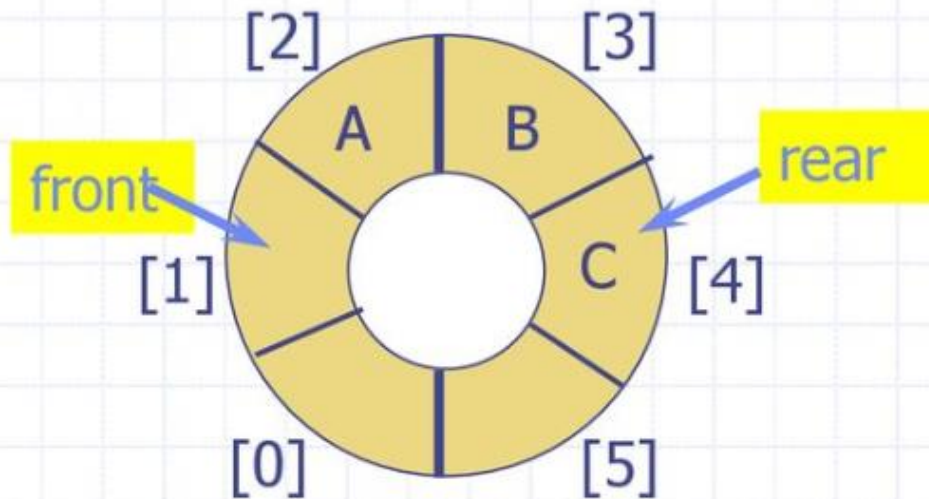
عنصر مورد نظر را در محل $queue[rear]$ قرار دهید



صف حلقوي

■ حذف کردن یک عنصر

front را در جهت عقربه های ساعت یک واحد افزایش دهید

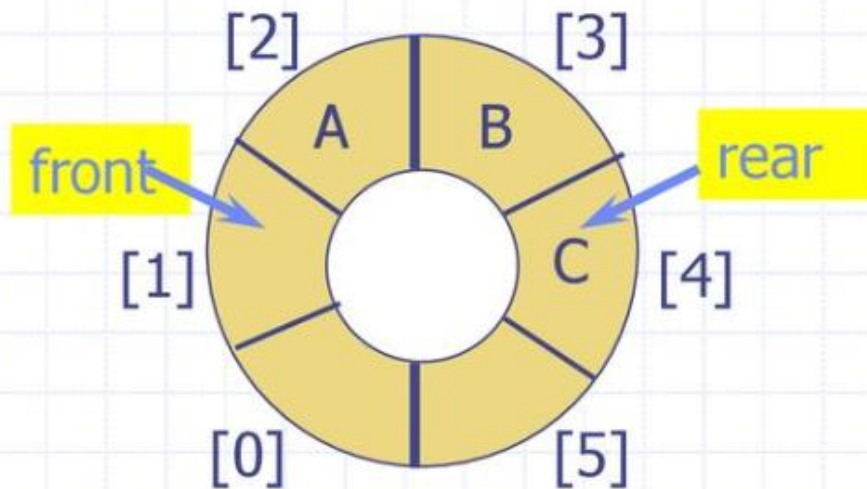


صف حلقوي

■ اضافه کردن به **rear** در جهت عقربه های ساعت

```
rear++;
```

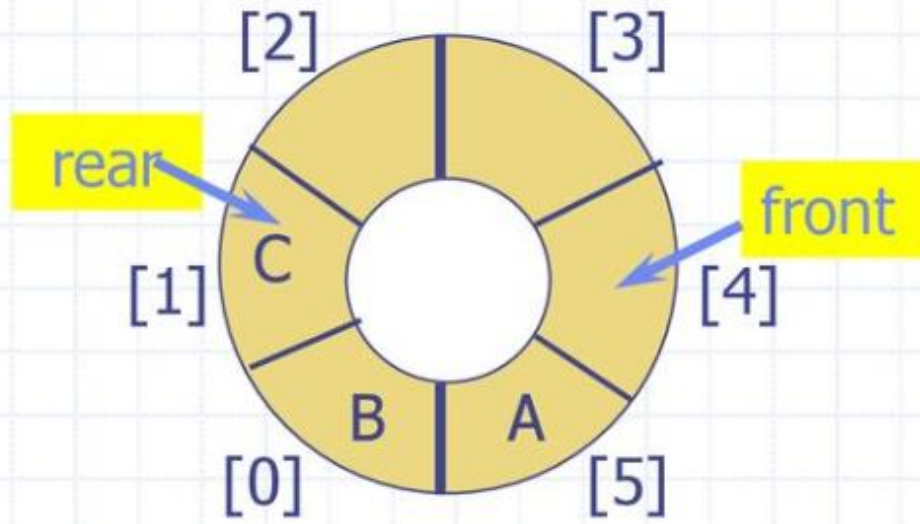
```
if (rear == capacity) rear = 0;
```



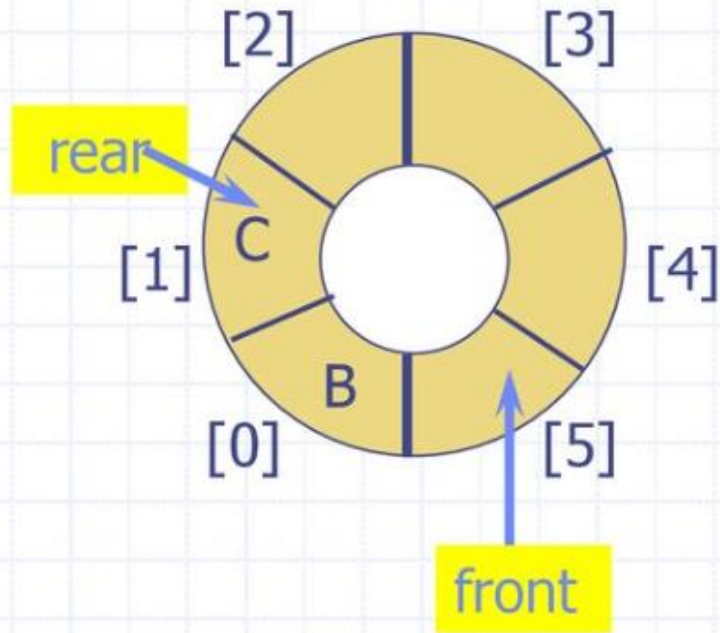
```
rear = (rear + 1) % capacity;
```

صف حلقي

■ صف خالي



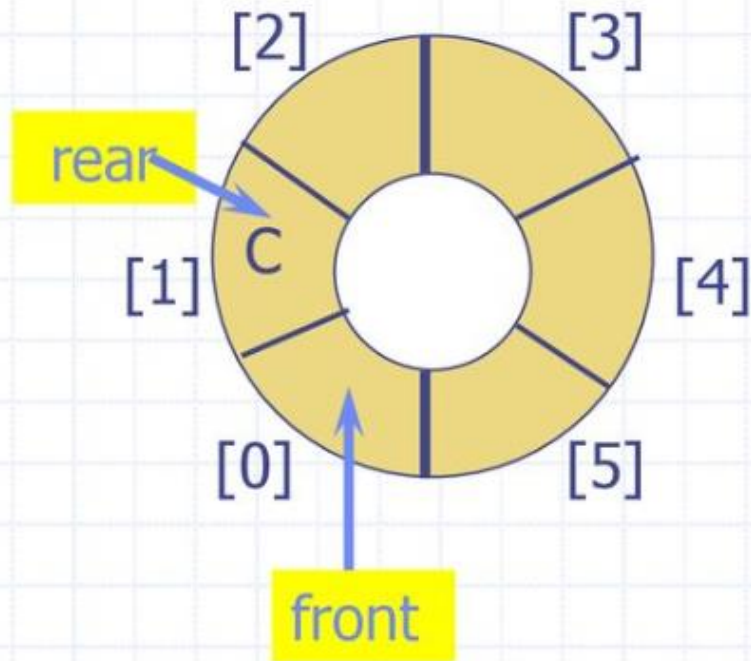
صف حلقي



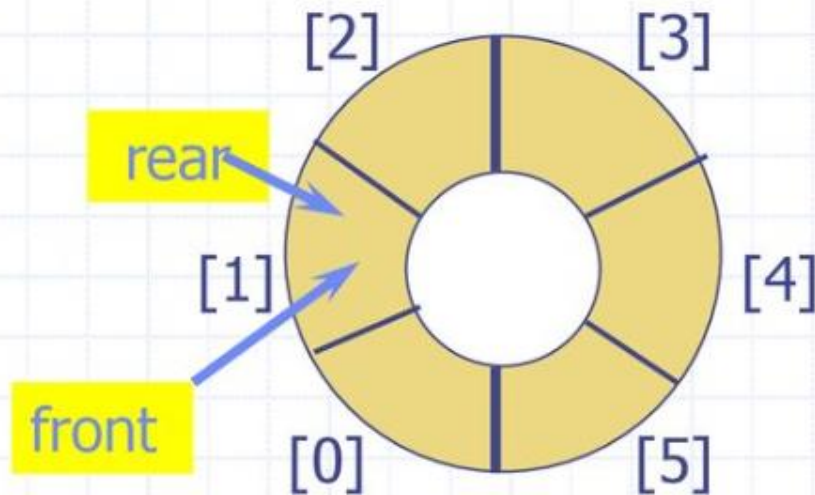
صف خالي

صف حلقي

صف خالي



صف حلقوي

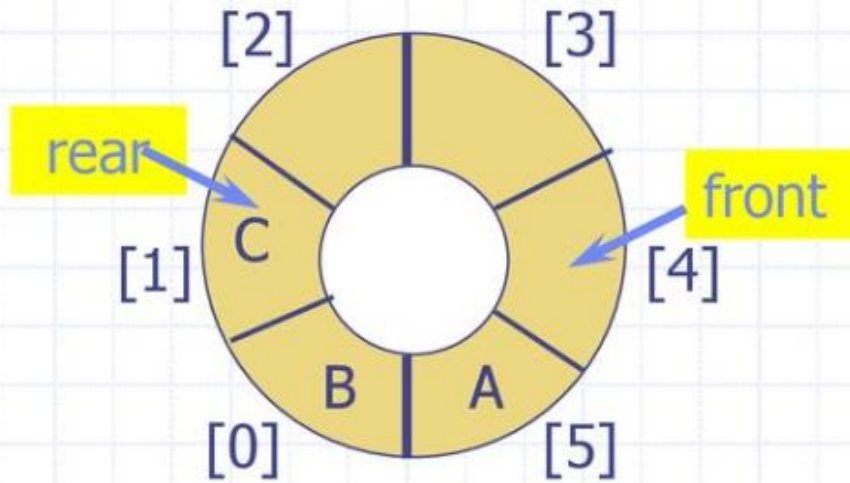


■ صف خالي

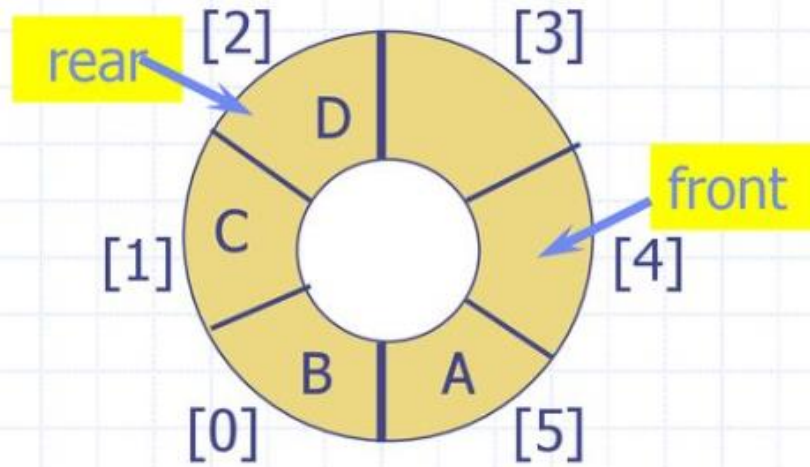
- پس از چند عمل حذف صف خالي مي شود، $front=rear$
- هنگامي که يك صف ايجاد مي شود خالي است
- پس مقدار اوليه $front = rear = 0$

صف حلقوي

صف پر

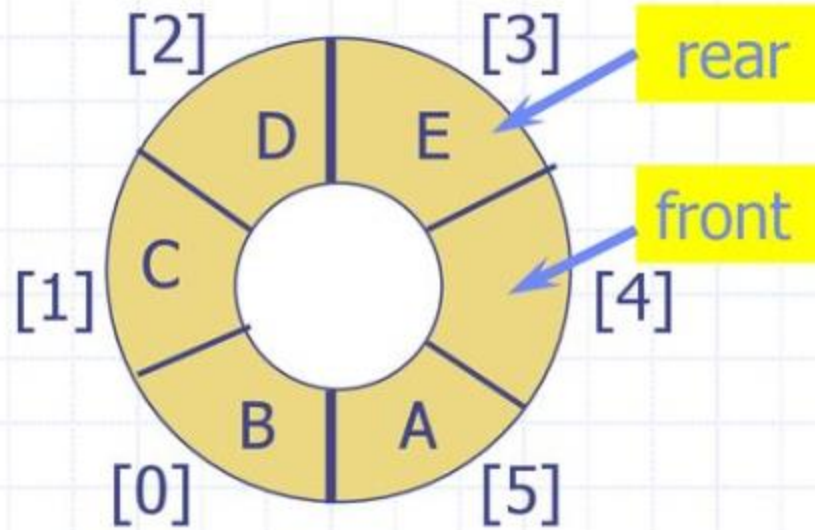


صف حلقوي



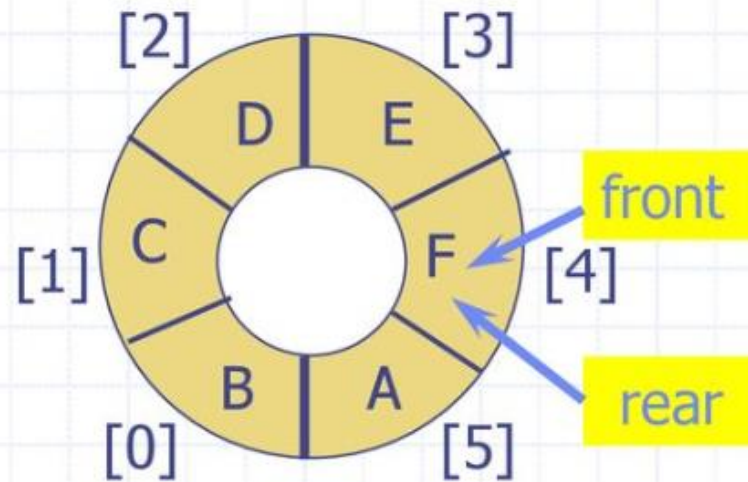
صف پر

صف حلقوي



صف پر

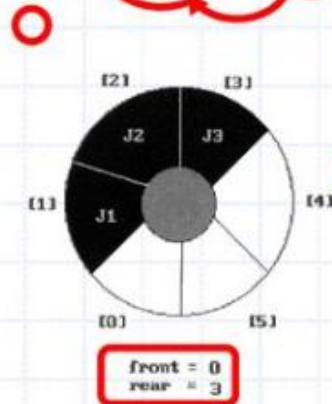
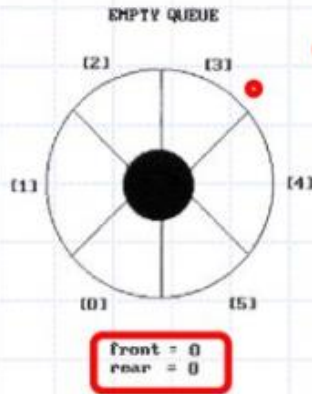
صف حلقوي



- هنگامي که یک سري عنصر به صف اضافه شود تا اين صف پر شود اين عمل باعث مي شود $front=rear$ شود
- بنابر اين نمي توان بين صف خالي و پر تمايز قائل شد

صف حلقوي

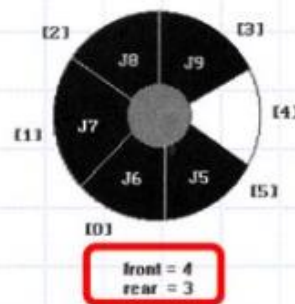
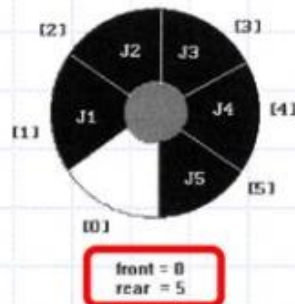
صف خالي
front=rear



صف غير خالي

صف پر
 $front = (rear + 1) \% \text{maxsize}$

FULL QUEUE FULL QUEUE



مشکل: یک فضای خالی باقی می ماند