

## مقدمه

در فرهنگ لغت، صف به معنای خط انتظار است.

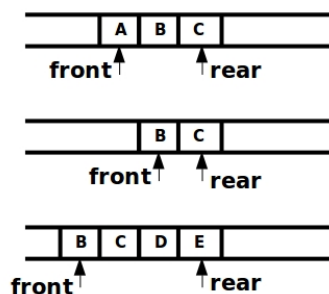
**مثال:** صفی از کارها در سیستم کامپیوتری که منتظرند از یک دستگاه خروجی مثلا چاپ گر استفاده کنند.

به طور رسمی، صف، مجموعه ای از عناصر مرتب است که هر کدام از عناصر از یک طرف به نام جلوی صف، از آن حذف می شود و از طرف دیگر به نام انتهای صف، در آن قرار می گیرند. به همین دلیل این ساختمان داده را از نوع *FIFO* (خروج به ترتیب ورود) توصیف می کنند.

## نوع داده انتزاعی صف

داده	مجموعه ی عناصر مرتب شده هستند. هر عنصر از یک طرف به نام جلوی صف، حذف و از طرف دیگر به نام انتهای صف، به آن اضافه می شوند.
عملیات	<ul style="list-style-type: none"><li>• ایجاد صف خالی</li><li>• تست خالی بودن صف</li><li>• افزودن عنصر به انتهای صف</li><li>• بازیابی عنصری از جلوی صف (ولی آن را از صف حذف نمی کند).</li><li>• حذف عنصر از جلوی صف</li></ul>

شکل زیر را در نظر بگیرید که مراحل مختلف حذف و اضافه عناصر را نشان می دهد.



متغیرهای *front* و *rear* به ترتیب آدرس های آخر و جلوی صف را نگه داری می کنند.

## پیاده سازی صف

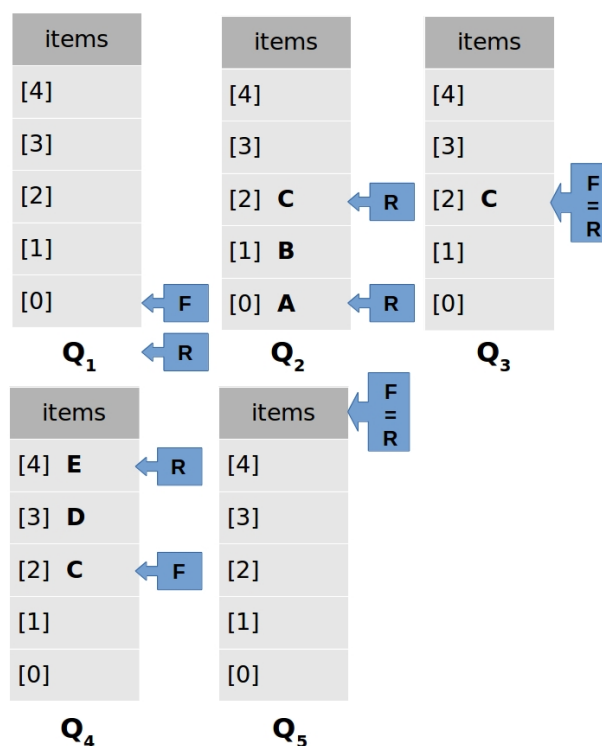
ساختمان داده صف را می توان با روش های

- خطی (آرایه)
- اشاره گر (لیست پیوندی)

پیاده سازی نمود. با عنایت به مشخصات هر کدام از آن ها، محدودیت هایی برای پیاده سازی ایجاد می شود. در ادامه به توضیح و ارائه راه حل برای برخی از این محدودیت ها می پردازیم.

### مشکلات پیاده سازی صف با آرایه

مشکل پیاده سازی صف با آرایه هنگام حذف و اضافه کردن عنصر جدید رخ می دهد. قسمت  $Q_4$  در شکل زیر را در نظر بگیرید:



از شرط  $(R)ear < (F)ront$  می توان برای تشخیص خالی بودن صف استفاده کرد. برنامه با توجه به این شرط، صف را پر تشخیص می دهد. در حالی که می توان در شکل دید که  $Q_4$  هنوز به اندازه ی 2 جای خالی گنجایش دارد. در این مرحله  $(R)ear$  مقدار 4 گرفته است. برای افزودن  $item$  جدید، باید مقدار آن 5 شود که خارج از حد آرایه است. بنابراین مشکل این است که با وجود خالی بودن صف، نمی توان عنصر جدید اضافه کرد.

جدول زیر مقادیر اشاره گر های  $(R)ear$  و  $(F)ront$  را دنبال می کند:

<b>Q1</b>	$(F)ront = 0$ $(R)ear = -1$
<b>Q2</b>	$(F)ront = 0$ $(R)ear = 2$
<b>Q3</b>	$(F)ront = (R)ear = 2$
<b>Q4</b>	$(F)ront = 2$ $(R)ear = 4$
<b>Q5</b>	$(F)ront = (R)ear = 5$

به هر حال، صف یک ساختمان داده است. مخاطب هر ساختمان داده ای، برنامه نویسان هستند. آن ها با در نظر گرفتن این نوع داده انتزاعی (کلاس) تلاش می کنند که تصور واضح تری از چیدمان داده ها و ارتباط آن ها با هم پیدا کنند. برای رفع مشکل مذکور، می توان صف را به صورت حلقوی تجسم کرد. یعنی متغیر های  $front$  و  $rear$  در صورت رسیدن به حدی مشخص، دوباره ریست شده و از 0 شروع کنند. در شکل زیر پیاده سازی صف حلقوی نشان داده شده است.

شکل

در این روش تشخیص خالی بودن صف است. شرط

$$\text{Rear} < \text{front}$$

اینجا صدق نمی کند. مثلاً شکل (ب) با وجود این که رابطه بالا صادق است ولی صف خالی نیست. بنابراین در این جا شده به صورت

$$\text{Rear} = \text{front}$$

بیان می شود.

پر بودن صف را نیز باید بتوان چک کرد. شرط بالا می تواند بدین منظور استفاده شود. بنابراین نمی توان پر بودن و خالی بودن صف را از هم تشخیص داد.

برای رفع این مشکل صف نیمه پر را ..... در نظر می گیریم.

شکل ص 100

وقتی یک محل خالی در آرایه وجود داشته باشد، صف را به عنوان صف پر در نظر می گیریم. شرط پر بودن به صورت زیر تغییر می کند:

$$\text{Size} = (\text{rear} - \text{front} + 1) \% \text{front}$$

در هر 2 نوع صف (خطی و حلقوی) ، تعداد عناصر موجود در صف از رابطه ی زیر بدست می آید:

$$\text{Rear} - \text{front} + 1$$

صف اولویت: در این نوع صف اضافه کردن عنصر جدید به هر ترتیبی امکان پذیر است ولی حذف یک عنصر از آن به صورت مرتب انجام می شود.

چند مثال را بررسی می کنیم.

مثال (صوف اولویت نزولی): در عمل حذف، بزرگ ترین عنصر صف حذف می شود.

مثال (صوف اولویت صعودی): در عمل حذف، کوچک ترین عنصر صف حذف می شود.

مثال: (صف سیستم عامل زمانی): آن کاری که مدت زمان بیشتری در صف انتظار، منتظر دریافت صنایع سیستم برده است.

اولویت حذف بالاتری نسبت به بقیه دارد و موقع تصمیم به حذف کردن یک عنصر زودتر از بقیه نوبت دهی از صف خارج می شود.

اصطلاحات:

Priority queue

Ascending priority queue

Descending priority queue

جمع بندی: مشکلات صف به طور کلی شامل موارد زیر می شود. اکثر این موارد هنگام برنامه نویسی پیش می آید. موارد دیگری نیز ممکن است پیش بینی شده اتفاق بیافتد.

- 1- محدودیت های آرایه: همه ی عناصر باید همگن باشند. (هم نوع هم اندازه)
  - 2- طول آرایه از پیش تعریف شده و ثابت است. با حذف تعداد زیادی عنصر [به خصوص در صف های اولویت که ترتیب حذف مستحق نیست] با آرایه اسپارس (خلوت) مواجه می شویم. اکثر مشکلات مذکور با بهبود روش های کدنویسی و یا تغییر در ساختمان داده، راه حل پیدا کرد:
  - تغییر فرم \_آرایه- تبدیل آن به صف حلقوی
  - استفاده از آرایه پویا
  - استفاده از لیست پیوندی
  - استفاده از سخت افزارهای قوی تر (حافظه بیشتر، پردازنده پر سرعت تر، )
  - تکنیک های مربوط به مدیریت بزرگ داده ها همچون پارتیشن بندی کردن
  - مربوط به قسمت مشکلات ص قبل
  - 3- به هنگام اضافه کردن عنصر در صف اولویت باید دوباره صف را مرتب سازی کنیم. مثلاً اگر اولویت صعودی بود، از کوچک تر به بزرگ تر مرتب کنیم. این طور نیست که عنصر جدید به طور خودکار (همانند صف معمولی) به آخر صف اضافه شود.
- کلاس پیاده سازی صف با لیست پیوندی ص 155.