



دانشکده علوم ریاضی و آمار



نیمسال اول ۱۴۰۰-۱۴۰۱

مدرس: دکتر مجتبی رفیعی

ساختمان داده‌ها و الگوریتم‌ها

جلسه ۲۵ ساختمان داده و الگوریتم‌ها

نگارنده: زهرا رشیدی

۷ آذر ۱۴۰۰

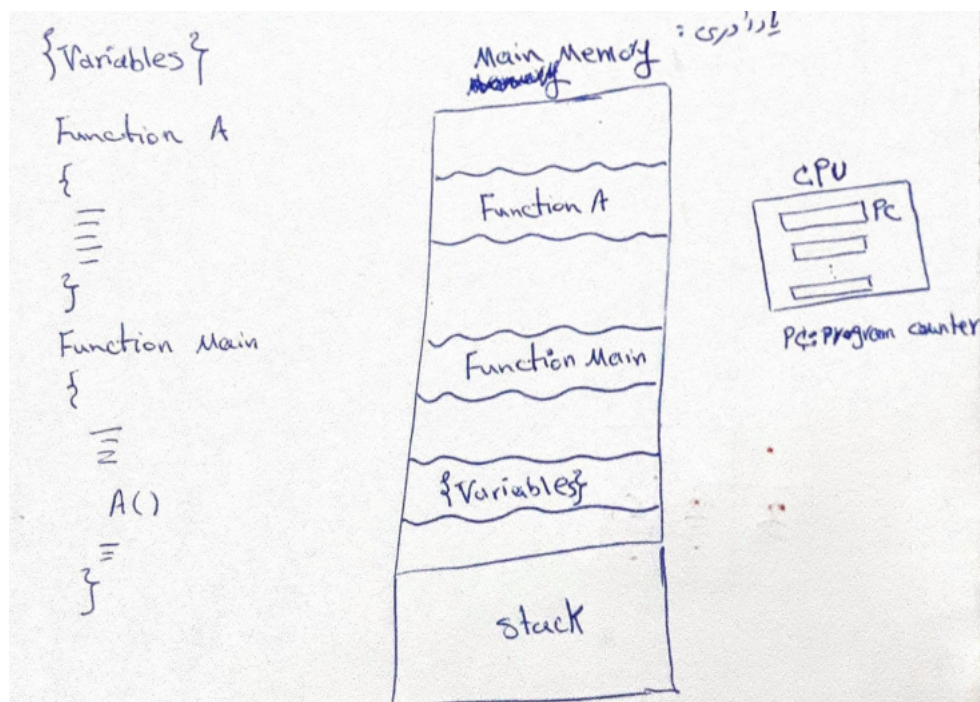
فهرست مطالب

۱	کاربردهای پشته
۱	۱.۱ فراخوانی تابع
۲	۲.۱ محاسبه دوره سهام روز i - ام
۳	۳.۱ مساله پارکینگ قطارها
۴	۲ داده ساختار صف (Queue)
۴	۱.۲ صف ساده
۵	۲.۲ صف حلقوی

۱ کاربردهای پشته

۱.۱ فراخوانی تابع

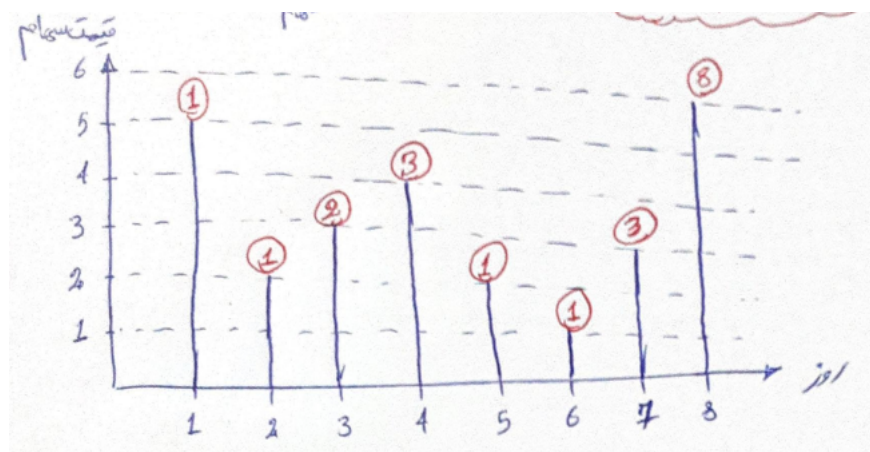
یکی از مهم ترین کاربردهای پشته است که در بخش های قبلی (الگوریتم های بازگشتی) تشریح کردیم.



شکل ۱: نمای کلی از برنامه، حافظه اصلی و واحد پردازش مرکزی

۲.۱ محاسبه دوره سهام روز i - ام

یک لیست از (روز و قیمت) سهام داده شده است. برای نمونه شکل زیر یک نمونه از این لیست را نشان می‌دهد.



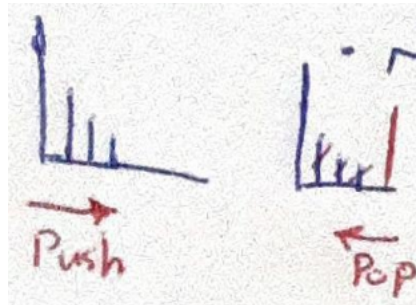
شکل ۲: لیست روز-قیمت سهام

تعریف دوره سهام روز i - ام : تعداد روزهای متوالی قبل از روز i - ام که قیمت سهام آنها در بازه کوچکتر یا مساوی روز i - ام هستند. لازم به ذکر است که خود روز i - ام را هم به این تعداد اضافه میکنیم. دوره سهام هر یک از روزها در شکل بالا مشخص شده است.

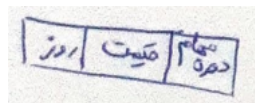
سوال: یک لیست به طول n از (روز و قیمت سهام) داده شده است. دوره سهام هر یک از روزهای لیست را محاسبه کنید.

- راه حل ۱: از روز مد نظر به عقب برمیگردیم و شرایط دوره سهام را چک میکنیم. در این راه حل، اگر قیمت ها به صورت صعودی باشد بدترین حالت اتفاق می افتد و پیچیدگی این راه حل $O(n^2)$ است.

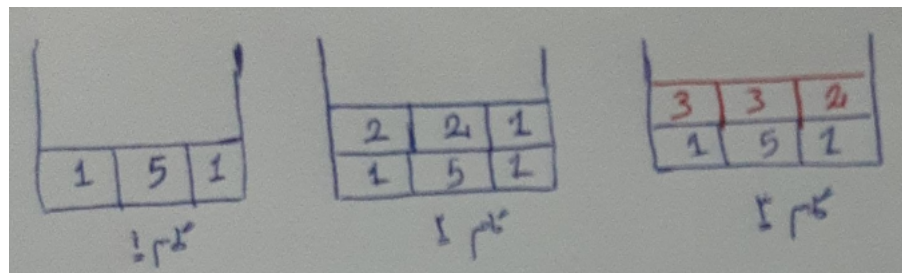
- راه حل ۲: استفاده از پشته و نگه داشتن عناصر به ترتیب نزولی در پشته.



شکل ۳: زمان اجرای عمل push و pop



شکل ۴: قالب هر عنصر در پشته



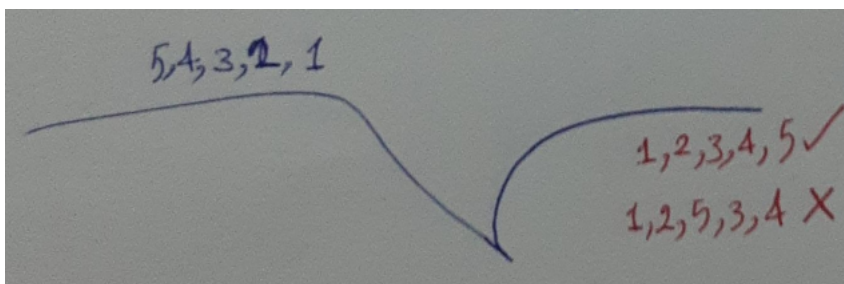
شکل ۵: سه گام از اجرای راه حل ۲

گام ۳: روز ۲ با دوره سهام ۱ از پشت خارج میشود و دوره سهام این روز به طور نهایی تعیین میشود و تاثیری هم در روزهای اتی ندارد. با این حال ۱ واحد برای روز سوم نیز لحاظ میشود (به خاطر pop).

پیچیدگی زمانی راه حل دوم: از آنجایی که هر عنصر تنها یکبار در پشته اضافه میشود و یکبار از پشته حذف میشود و دوره سهام آن روز تعیین میشود، هزینه کلی $O(n)$ است.

۳.۱ مساله پارکینگ قطارها

یک لیست از ترتیب ورود قطارها به پارکینگ داریم هر قطاری که وارد پارکینگ میشود امکان برگشت به ورودی پارکینگ را ندارد. از بین کسانی که در پارکینگ قرار دارند، کسی که زودتر وارد پارکینگ شده، امکان خروج دارد.



شکل ۶: یک نمونه از خروج مجاز/غیر مجاز

صورت مساله پارکینگ قطارها: با توجه به قواعد و لیست وارد شده برای ترتیب ورود قطارها، آیا لیست تعیین شده برای خروجی قطارها از پارکینگ معتبر است؟

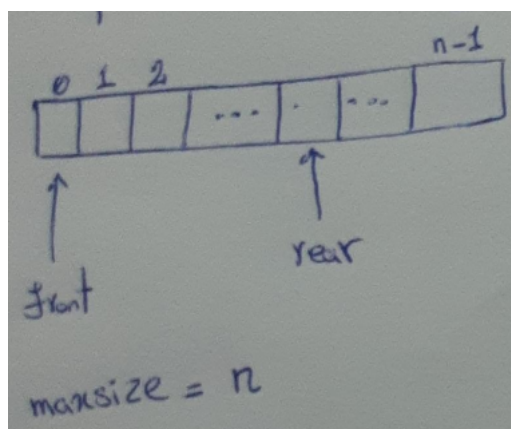
نکته: داده ساختار پشته را می توان با استفاده از لیست ساده و یا لیست پیوندی پیاده سازی کرد.

۲ داده ساختار صف (Queue)

یک داده ساختار برای نگهداری مجموعه های پویاست که عناصر آن در یک ترتیب خطی قرار گرفته اند و برای حذف عناصر در آن از سیاست "اولین ورودی، اولین خروجی" (FIFO=First In First Out) استفاده می شود. در ادامه از دو نمایش گرافیکی برای داده ساختار صف معرفی می کنیم:

۱.۲ صف ساده

شکل زیر نمایش گرافیکی این ساختار پیشنهادی را نشان می دهد:



شکل ۷: نمایش گرافیکی صف ساده

نکات زیر برای این نحو نمایش قابل بیان است:

- front به ابتدای صف اشاره می کند.
- rear به جایی اشاره می کند که داده جدید باید درج شود (انتهای صف).
- اگر $front=rear$ باشد به این معناست که صف Q خالی است.

- اگر $rear = maxsize$ باشد به این معناست که صف Q پر است.

عملیات قابل تعریف روی صف ساده به قرار زیر است:

- **Queue-EnQueue(Q,x)**: یک پرسمان بروزسانی است که عنصر x را به انتهای صف اضافه میکند. لازم به ذکر است اگر صف پر باشد، پیام بر "overflow" گرداننده میشود.

```
1: if ( $Q.rear == Q.maxsize$ ) then
2:   return "overflow"
3: else
4:    $Q[Q.rear] = x$ 
5:    $Q.rear = Q.rear + 1$ 
```

- **Queue-DeQueue(Q)**: یک پرسمان بروزسانی است که عنصر ابتدایی صف را حذف و برمیگرداند. لازم به ذکر است که اگر صف خالی باشد، پیام برگرداننده "underflow" می شود.

```
1: if ( $front == rear$ ) then
2:   return "underflow"
3: else
4:    $x = Q[Q.front]$ 
5:    $Q.front = Q.front + 1$ 
```

مشکل صف ساده: پس از انجام تعدادی عمل حذف و درج در صف، با اینکه آرایه فضای آزاد دارد، امکان درج عنصر جدید را نخواهیم داشت.

۲.۲ صف حلقوی

این نوع صف در جلسه بعد تشریح می شود.