



نیمسال اول ۱۴۰۰-۱۴۰۱

مدرس: دکتر مجتبی رفیعی

ساختمان داده‌ها و الگوریتم‌ها

جلسه ۲۴ ساختمان داده و الگوریتم‌ها

نگارنده: فرزانه مولایی

۷ آذر ۱۴۰۰

فهرست مطالب

۱	شهود پیاده سازی لیست پیوندی با استفاده از آرایه
۲	۱.۱ پیاده سازی با یک آرایه
۲	۲.۱ پیاده سازی با چند آرایه
۳	۲ داده ساختار پشته (Stack)
۳	۳ عملیات روی پشته

۱ شهود پیاده سازی لیست پیوندی با استفاده از آرایه

- یادآوری: لیست پیوندی یک ترتیب خطی منطبق با یک سری اشاره گر است. زمانی که یک آرایه تعریف می کنیم منظور یک ترتیب خطی منطبق با یک سری اندیس می باشد.

<i>prev</i>	<i>element</i>	<i>next</i>
-------------	----------------	-------------

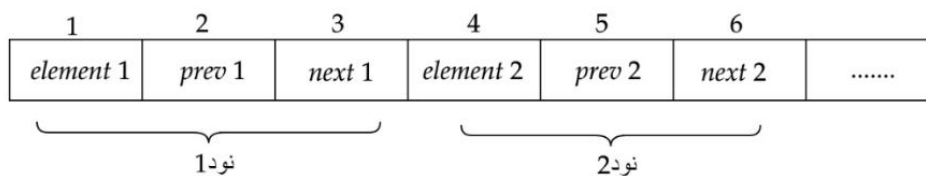
شکل ۱: قالب لیست پیوندی دو طرفه

در ادامه سعی داریم پیاده سازی لیست پیوندی را با یک آرایه و چند آرایه شرح دهیم.

۱.۱ پیاده سازی با یک آرایه

۱- نمایش null با مقدار صفر است.

۲- $prev_i$ و $next_i$ حاوی اندیس های مناسبی از آرایه است.



شکل ۲: پیاده سازی لیست پیوندی دوطرفه با یک آرایه .

۲.۱ پیاده سازی با چند آرایه

	1	2	3	
<i>prev</i>			
<i>element</i>			
<i>next</i>			

شکل ۳: پیاده سازی لیست پیوندی دوطرفه با چند آرایه

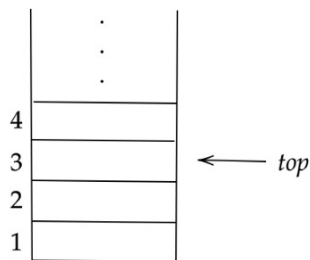
چالش ها: برای درج (نیازمند پیدا کردن خانه خالی برای درج) / حذف (نیازمند بروزرسانی خانه ها برای خالی بودن) / جستجو (نیازمند اشاره به نود اول) را بررسی کنید.

۲ داده ساختار پشته (Stack)

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 "آخرین ورودی، اولین خروجی" استفاده می شود. اشاره گر (top) نشان دهنده اندیس بالای پشته می باشد.

نکته

اگر $top=0$ باشد یعنی پشته خالی است.



شکل ۴: نمایش گرافیکی پشته

۱.۲ عملیات روی پشته

۱. $Stack_Push(S, x)$: یک پرسمان بروزسانی است که عنصر x را به بالای پشته S اضافه می کند .

$Stack_Push(S, x)$
1. $S.top = S.top + 1$
2. $S[S.top] = x$

پیچیدگی زمانی الگوریتم فوق $O(1)$ است .

۲. $Stack_Pop(S)$: یک پرسمان بروزسانی است که عنصر بالای پشته را حذف و برمیگرداند . لازم به ذکر است که اگر پشته تهی باشد پیام "underflow" برگردانده می شود .

```
1: if (S.top == 0) then
2:   return "underflow"
3: else
4:   return S.top = S.top - 1
5:   return S[S.top + 1]
```

پیچیدگی زمانی الگوریتم فوق $O(1)$ است .

۳. **Stack-Empty(S)**: یک پرسمان بازیابی است که تعیین می کند که آیا پشته خالی است یا نه .

```
1: if ( $S.top == 0$ ) then  
2:   return True  
3: else  
4:   return False
```

۴. پیچیدگی الگوریتم فوق $O(1)$ است .

۵. **Stack-Top(S)**: یک پرسمان بازیابی است که عنصر بالایی پشته را بدون تغییر در پشته بر میگرداند . لازم به ذکر است که اگر پشته تهی باشد پیام "underflow" را برمی گرداند .

```
1: if ( $S.top == 0$ ) then  
2:   return "underflow"  
3: else  
4:   return  $S[S.top]$ 
```

پیچیدگی زمانی الگوریتم فوق $O(1)$ است .