



FARZIN HAMZEHI

FRONT-END DEVELOPER



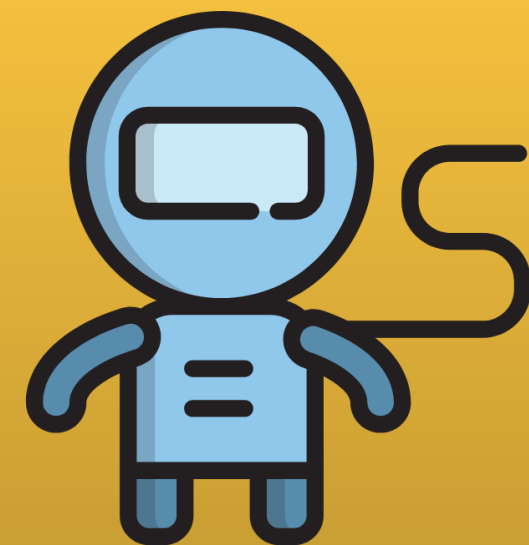
جبررسی داده ساختار

STACK



در جاوا اسکریپت

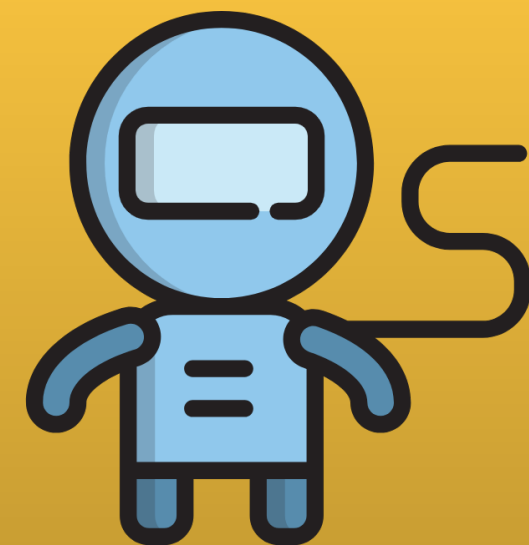




تعریف :

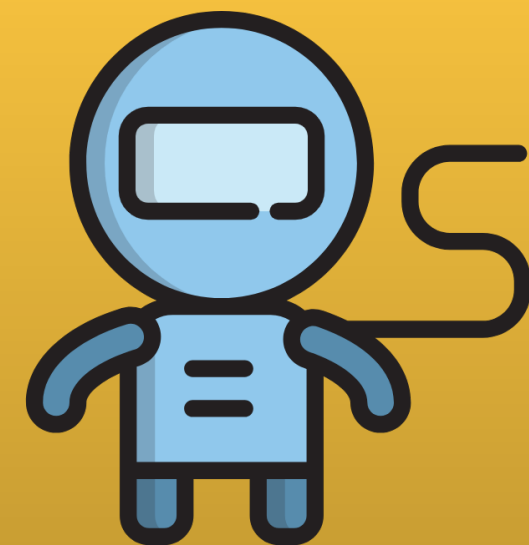
پشته یک داده ساختار مهم در الگوریتمه که
بر مبنای لیفو عمل میکنه یعنی آخرین
عنصری که وارد میشه اولین عنصریه که
خارج میشه.

الگوریتم مهمی که خیلی پر کاربرد و پای ثابت مصاحبه ها هستش



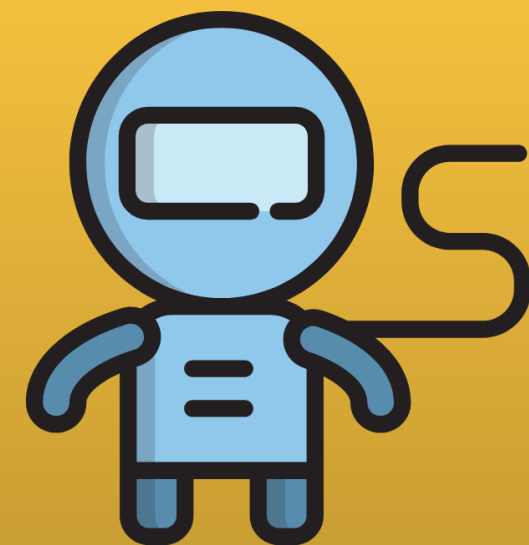
پشته یک داده ساختار انتزاعی

پشته یک داده ساختار انتزاعی مدل سازی میشه
و به صورت فیزیکی در حافظه فیزیکی پیاده
سازی نمیشه پس به این دلیل میتوان با آرایه و
لیست پیوندی و ... پشته رو پیاده سازی کرد.

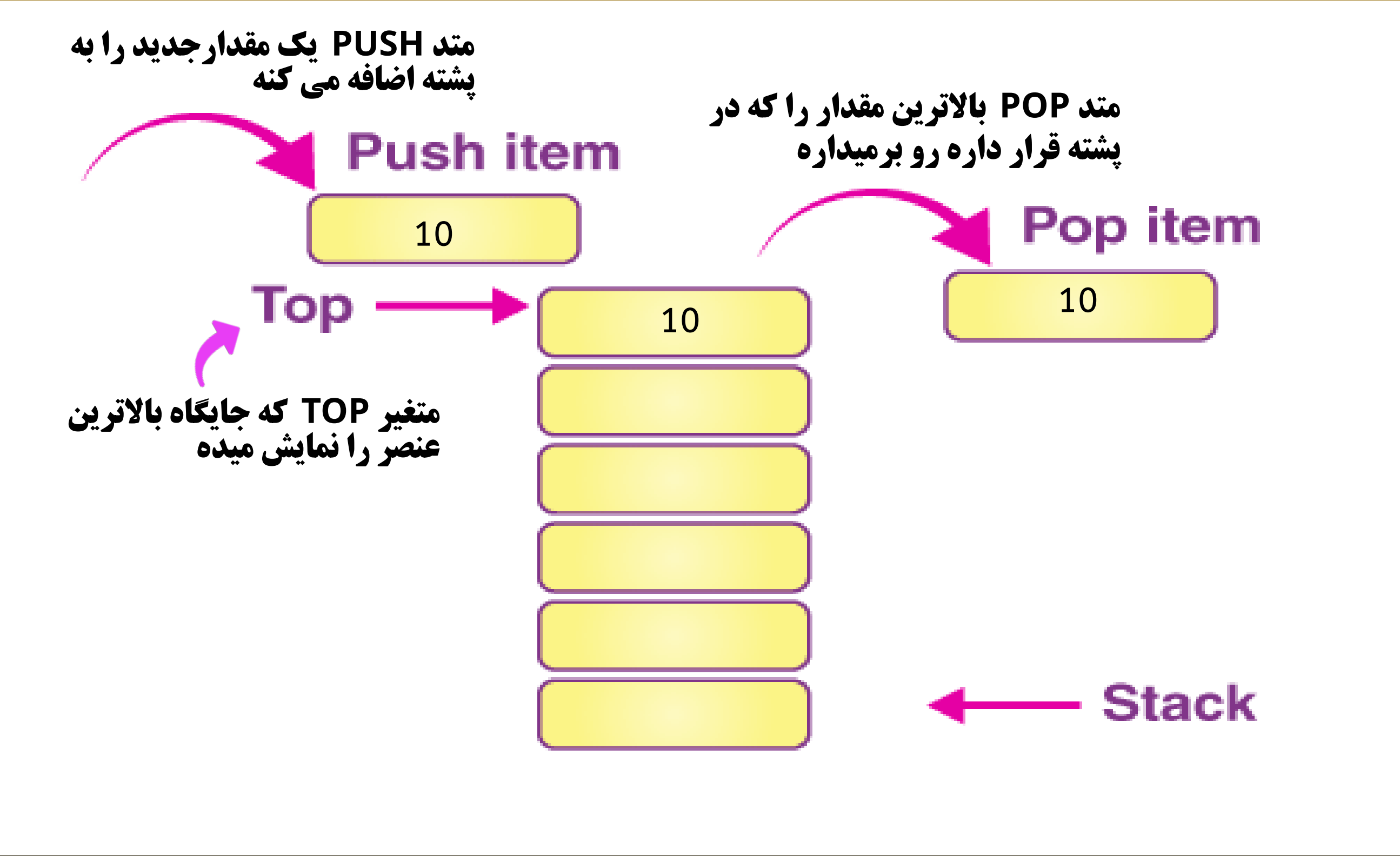


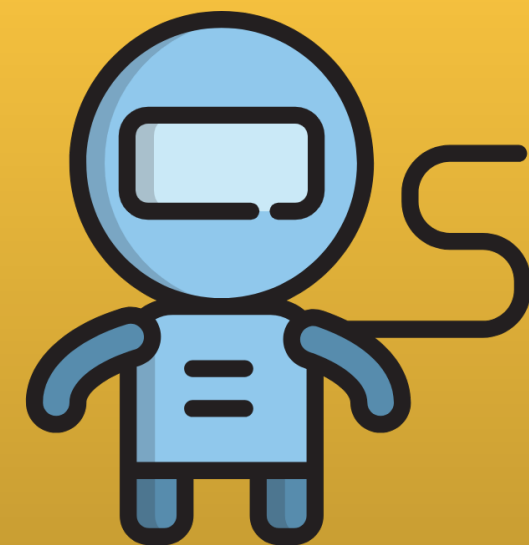
توضیح الگوریتم

۱. در پشته برای اضافه کردن یا برداشتن هر عنصر ما باید جایگاه بالاترین عنصر رو بدونیم , بیشتر اوقات این مقدار در متغیری به اسم TOP ذخیره می کنیم.
۲. در ابتدا که پشته خالیه و هیچ عنصری داخل آن وجود نداره مقدار TOP برابر ۱- قرار میدیم.
۳. پشته دو متد اصلی به نام های PUSH که برای اضافه کردن مقدار به پشته و POP که برای برداشتن مقدار از پشته است.
۴. مقدار TOP در PUSH یکی اضافه میشه ولی در POP یکی از آن کم می کنیم.



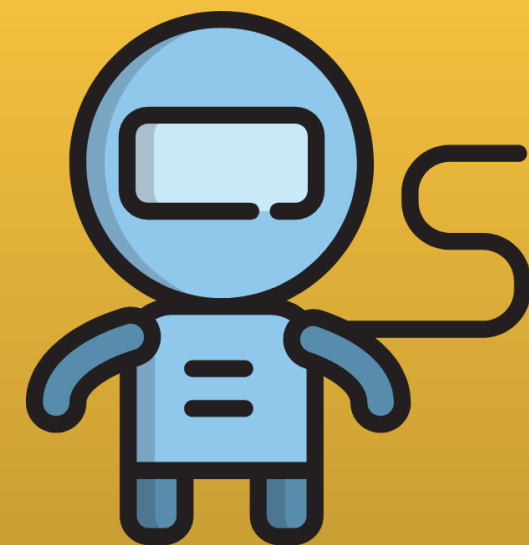
توضیح پشته به صورت شهودی





نکات ضعف

۱. حداکثر تعداد عناصری که یک پشته می‌تونه نگه داری بکنه محدوده و ما در پشته مشکل اندازه محدود رو داریم .
- ۲ . ما در پشته فقط میتونیم به عنصر آخر دسترسی پیدا کنیم و عملاً مفهوم پیمایش کردن در پشته جایگاهی نداره.
- ۳ . در برخی پیاده سازی ها ممکنه باعث بشه که در حافظه مشکلاتی ایجاد بشه.

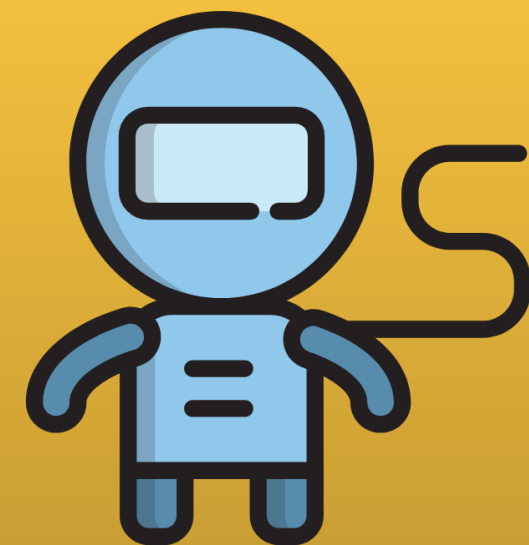


نکات قوت

۱. همان طور که قبلا گفته شد الگوریتم پشته یک نوع از الگوریتم های انتزاعی است پس میتوان با آرایه , لیست پیوندی و ... برای پیاده سازی این الگوریتم استفاده کرد.

۲. این الگوریتم در زبان های مختلف پیاده سازی ساده ای داره.

۳. این الگوریتم نسبت به بقیه الگوریتم ها مانند صف سرعت بیشتری داره.



مثال جا کد

```
1 class Stack {
2   constructor() {
3     this.items = [];
4   }
5
6   push(element) {
7     this.items.push(element);
8   }
9
10  pop() {
11    if (this.isEmpty()) {
12      throw new Error("Pop from an empty stack");
13    }
14    return this.items.pop();
15  }
16
17  isEmpty() {
18    return this.items.length === 0;
19  }
20
21  size() {
22    return this.items.length;
23  }
24 }
25
26 const myStack = new Stack();
27
28 myStack.push(10);
29 myStack.push(20);
30 myStack.push(30);
31
32 console.log("Stack size:", myStack.size()); // 3
33
34 console.log("Pop element:", myStack.pop()); // 30
35
36 console.log("Is the stack empty?", myStack.isEmpty()); // false
37
```

← ساخت کلاس STACK

← متد PUSH که ELEMENT رو به پشته اضافه می کنه

← متد POP که ELEMENT رو در صورت خالی نبودن پشته بر میگرددونه.

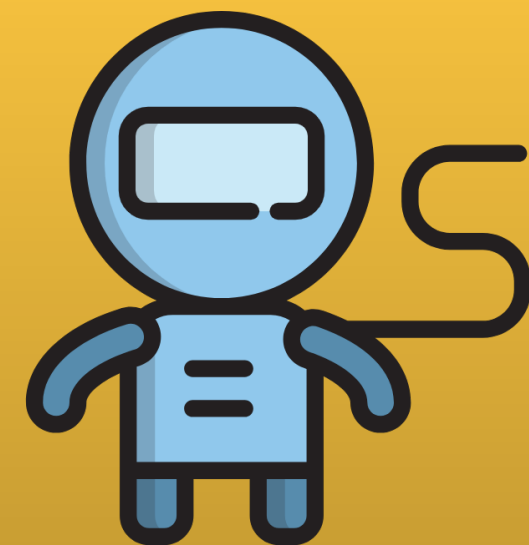
← متد IS EMPTY مشخص میکنه که آیا پشته خالیه یا نه

← متد SIZE اندازه پشته رو به ما برمیگردونه

← ساختن یک شی جدید از STACK

← اضافه کردن مقدار جدید توسط متد PUSH





نکات مهم

۱. زمان اجرای هر کدان از عملیات $PUSH$, POP در پشته ثابت است و از نوع $O(1)$ است.

۲. پشته بیشتر در فراخوانی توابع بازگشتی کاربرد دارد، هنگامی که یک تابع فراخوانی میشه اطلاعات و ترتیب اجرا توابع در پشته ذخیره میشه.

۳. در بسیاری از الگوریتم‌ها مانند الگوریتم DFS (جستجوی عمق اول) و حل مسائل گراف‌ی، پشته به عنوان یک ابزار مؤثر برای ذخیره و بازگشت به اطلاعات استفاده می‌شود.





ممنون از وقتی که گذاشتین!

