

به نام خدا

ساختمان داده ها و الگوریتم ها

محمد مهدی گیلانیان صادقی

دانشگاه آزاد اسلامی واحد قزوین

نیمسال دوم ۱۴۰۱-۱۴۰۲



• پشته (Stack)

مجموعه ای از عناصر مرتب که از یک طرف (بالای پشته) قابل دستیابی هستند. چون در پشته آخرین عنصر ورودی، اولین عنصری است که از پشته خارج می شود آنرا ساختمان داده Last In First Out (LIFO) می نامند.

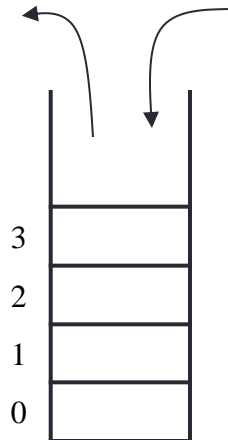
پیاده سازی پشته:

- با استفاده از آرایه ها
- با استفاده از لیست های پیوندی

پیاده سازی پشته با استفاده از آرایه ها:

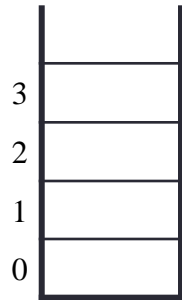
عملیات اصلی :

- ایجاد پشته خالی
- بررسی خالی بودن پشته
- بررسی پر بودن پشته
- افزودن عنصری به بالای پشته
- حذف عنصری از بالای پشته
- بازیابی عنصری بالای پشته





مثال: top به عنصر بالای پشته اشاره می کند. در ابتدا $top = -1$ است.

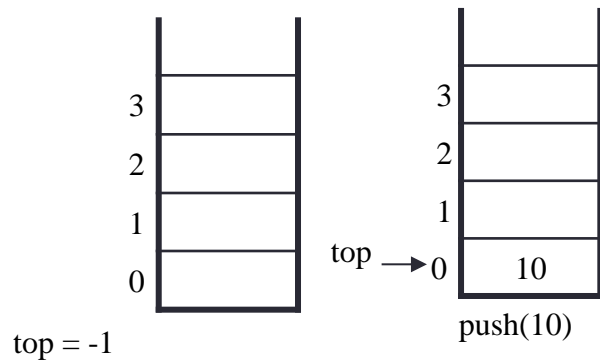


$top = -1$

نمونه ای از اعمال حذف و اضافه در پشته



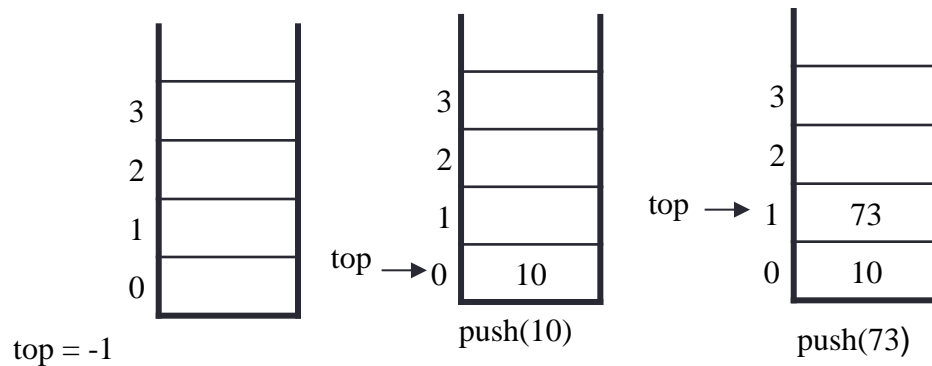
مثال: top به عنصر بالای پشته اشاره می کند. در ابتدا $top = -1$ است.



نمونه ای از اعمال حذف و اضافه در پشته



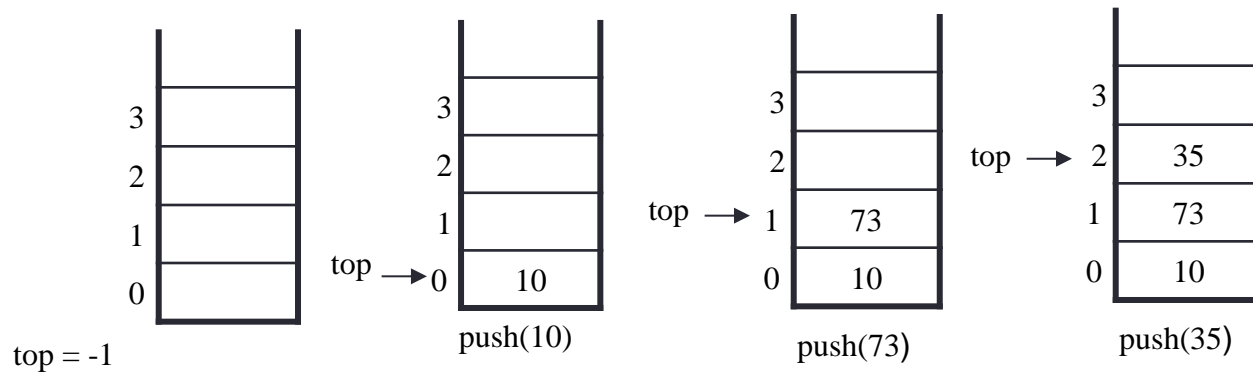
مثال: top به عنصر بالای پشته اشاره می کند. در ابتدا $top = -1$ است.



نمونه ای از اعمال حذف و اضافه در پشته



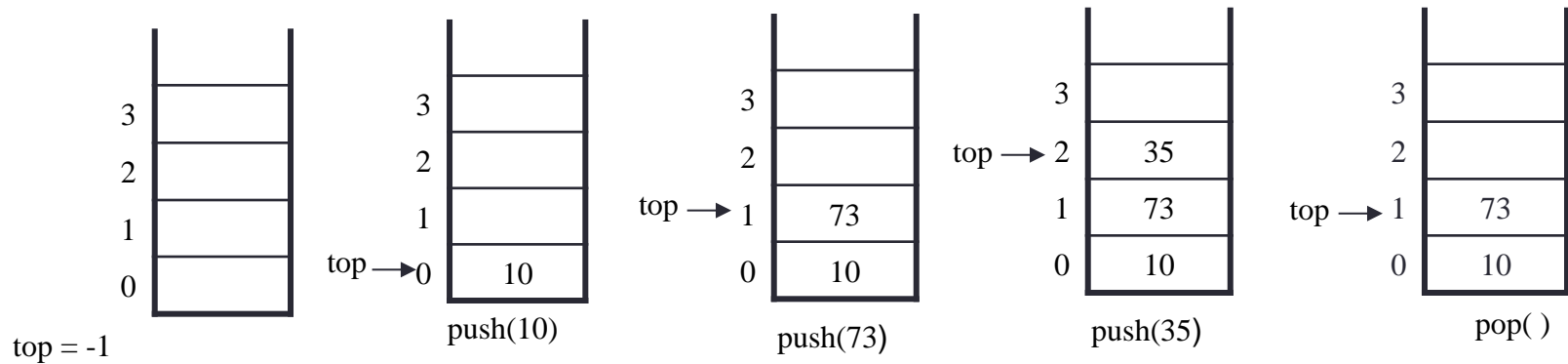
مثال: top به عنصر بالای پشته اشاره می کند. در ابتدا $top = -1$ است.



نمونه ای از اعمال حذف و اضافه در پشته



مثال: top به عنصر بالای پشته اشاره می کند. در ابتدا $top = -1$ است.



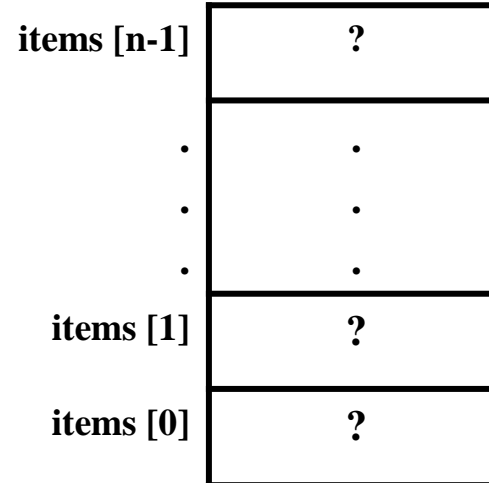
نمونه ای از اعمال حذف و اضافه در پشته



```
# define n 5
class stack {
    public:
        stack();
        int empty();
        int full();
        void push(int x);
        int pop();
        // other member functions
    private:
        int top;
        int items[n];
};
```

```
stack s;
```

پیاده سازی کلاس پشته:



top = -1

نمونه ای از پشته با آرایه



ساختمان داده ها و الگوریتم ها

صفحه ۹

```
stack::stack()
{
    top= -1;
}
```

پیاده سازی عمل ایجاد پشته



ساختمان داده ها و الگوریتم ها

صفحه ۱۰

پیاده سازی عمل ایجاد پشته

```
stack::stack()
{
    top= -1;
}
```

پیاده سازی عمل بررسی خالی بودن پشته

```
int stack::empty()
{
    if (top == -1)
        return 1;
    return 0;
}
```

معادل return (top == -1);



ساختمان داده ها و الگوریتم ها

صفحه ۱۱

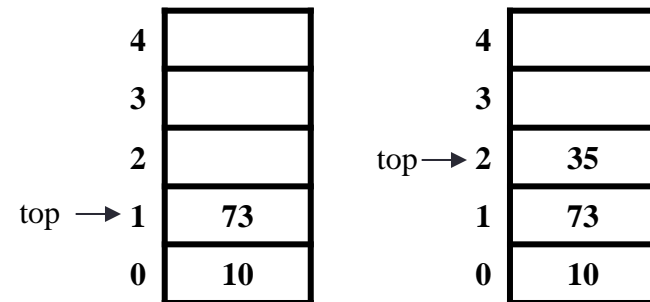
پیاده سازی عمل بررسی پر بودن پشته

```
int stack::full()
{
    if (top == n-1)
        return 1;
    return 0;
}
```



```
void stack::push(int x)
{
    if (full ())
    {
        cout<<"stack is full";
        exit(1);
    }
    else
        items[++top]=x;
}
```

پیاده سازی عمل افزودن به پشته



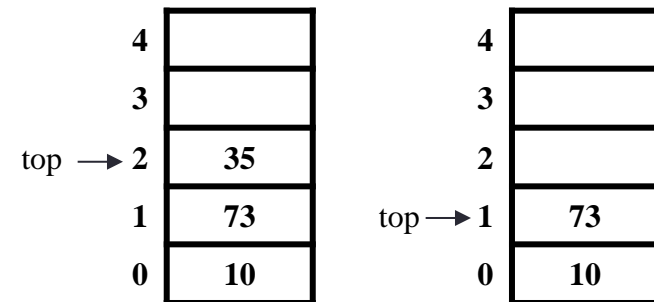
```
stack s;
int x;
cin>>x;
s.push(x);
```

x=35



```
int stack::pop()
{
    if (empty ())
    {
        cout<<"stack is empty";
        exit(1);
    }
    else
        return items[top--];
}
```

پیاده سازی عمل حذف از پشته



```
stack s;
int y;
y=s.pop();
```

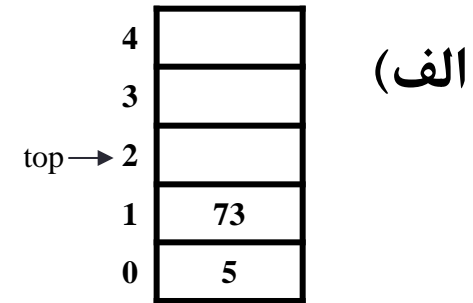
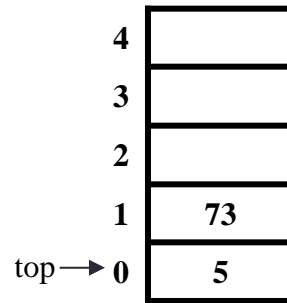
y=35



ساختمان داده ها و الگوریتم ها

صفحه ۱۴

تمرین ۱: زیر برنامه های `empty`, `full`, `push` و `pop` را برای پشته های زیر بنویسید؟



تمرین ۲: زیر برنامه ای به نام `prints` بنویسید که محتویات پشته را چاپ کند؟

تمرین ۳: زیر برنامه ای به نام `tops` بنویسید که مقدار بالای پشته را برگرداند؟ (از پشته حذف نمی کند).



پایان