پشته (Stack)

قاسم مهدور (mahdevar@ibb.ut.ac.ir)

نوع داده پشته (Stack)

- تعریف:

ساختمان داده خطی می‌باشد که افزودن به آن و یا کاستن از آن فقط از یک سمت ممکن است.

مثال:

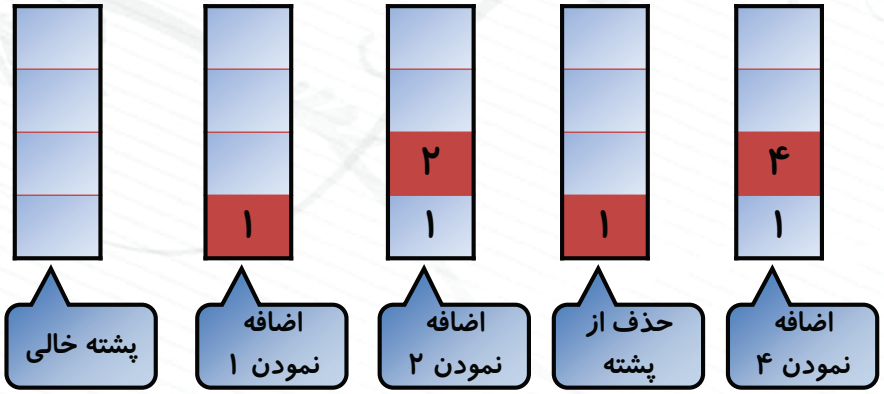
$X_0, X_1, \ldots, X_n$

- در پشته فوق عنصر X_n در بالا قرار دارد. می‌توان این عنصر را حذف و یا عنصر جدیدی بعد از آن اضافه نمود.
- اندازه پشته خالی 0 است.
- موقعیت عضو i ام، یعنی X_i همان i است.
- به اولین عنصر پشته bottom و آخرین آنها top می‌گوییم.

فهرست مطالب

دسترسی به اشیاء داخل صف و پشته

- در پشته آخرین عنصر ورودی (Last In) اولین عنصر خروجی (First Out) می‌باشد. به یک چنین دسترسی، دسترسی آخرین ورودی اولین خروجی LIFO(Last In First Out) می‌گویند.



اشیا داخل پشته

- یک پشته از
  - اعداد صحیح، اعشاری، دودویی، ...
  - رشته‌های حرفی،
  - انواع داده‌های دیگر(؟)

  را می‌توان ساخت.

مثال

- یک پشته از اعداد صحیح

  1, 2, 3, 4
- نتیجه اجرای برخی از اعمال روی این پشته

  Top() ⟶ 4

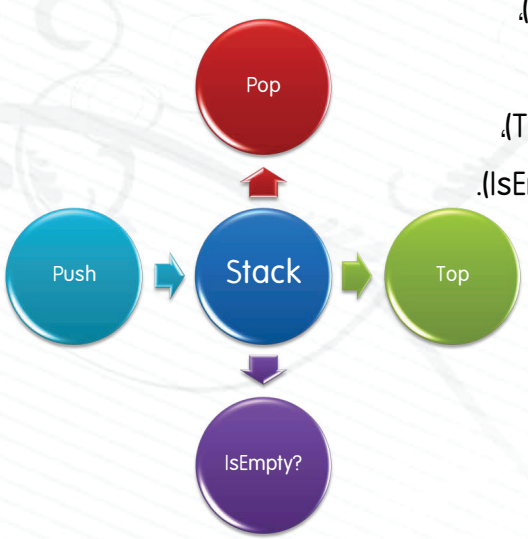
  Pop() ⟶ 4

  Push(5) ⟶ 1, 2, 3, 5

  IsEmpty() ⟶ FALSE

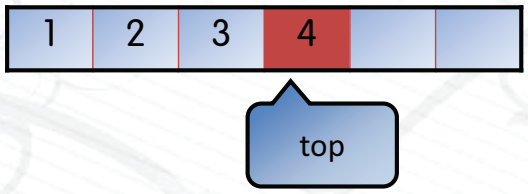
اعمالی که می‌توان روی پشته انجام داد

1. ایجاد (Create)،
2. اضافه نمودن یک عضو (Push)،
3. حذف یک عضو (Pop)،
4. دریافت مقدار بالای پشته (Top)،
5. تعیین خالی بودن پشته (IsEmpty).

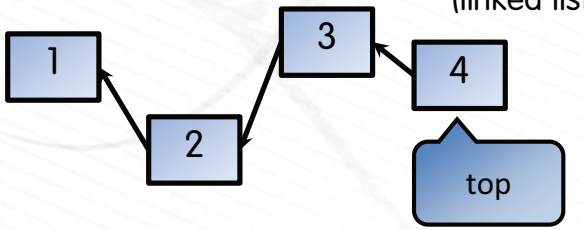


روش‌های پیاده سازی پشته

- آرایه (array)،



- فهرست پیوندی(linked list)



پیاده سازی با استفاده از آرایه

- پیاده سازی پشته با استفاده از آرایه
- چگونگی اضافه نمودن به پشته
- نحوه حذف از پشته

پیاده سازی ساختار پشته در C

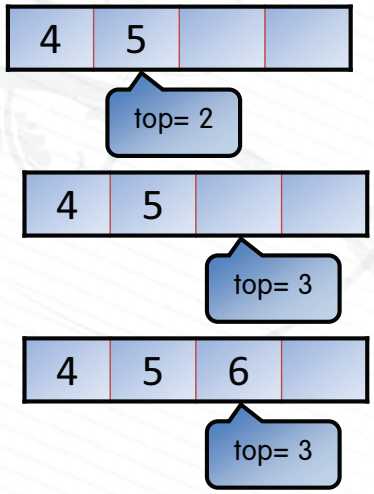
```
#define MAX 100
struct Stack
{
    int TopOfStack;
    int Data[MAX];
};

void main()
{
    Stack A;
    A->TopOfStack=-1;
    Push(A, 1);
}
```


پیاده سازی پشته با استفاده از آرایه

- برای انجام این کار به یک آرایه و یک متغیر جهت دانستن اندیس عنصر top (که همان تعداد اعضای داخل پشته است) نیاز داریم.
- ویژگی‌های این روش:
  - پر کاربردتر از روش دیگر (فهرست پیوندی) است؛ زیرا سرعت بالای دارد(اضافه و حذف نمودن فقط در انتها رخ می‌دهد)، و
  - باید بیشترین اندازه پشته معلوم باشد!
- با استفاده از باز تخصیص نمودن حافظه (مثلاً تابع realloc در C) می‌توان نقص را مرتفع نمود. البته این گزاره همواره صحیح نیست! چرا؟؟؟

چگونگی اضافه نمودن به پشته (Push)

- برای اضافه نمودن عنصر جدید باید
  - متغیر top اضافه شود، و
  - مقدار جدید در مکان top نوشته شود.



الگوریتم اضافه نمودن به پشته

- void Push(Stack *A, int d)

```
{
  A->top++;
  A->Data[top] = d;
}
```

- آیا ممکن است این الگوریتم با خطا مواجه شود؟ اگرجواب بلی است چگونه می‌توان آن را حل نمود؟

نحوه حذف عنصر top (Pop)

- void Pop(Stack *A)

```
{
  A->top--;
}
```

- آیا ممکن است این الگوریتم درست عمل نکند؟ اگرجواب بلی است چگونه می‌توان آن را تصحیح نمود؟

یافتن مقدار top (Top)

- int Top(Stack *A)

```
{
  return A->Data[top];
}
```


چگونگی تشخیص خالی بودن پشته (IsEmpty)

- در ابتدا مقدار top را برابر 1- قرار می‌دهیم؛ از آنجا که
  - با اضافه نمودن عنصر جدید این متغیر بعلاوه یک می‌شود و
  - با حذف عنصر این متغیر یک واحد کاهش می‌یابد
- بنابراین هرگاه این متغیر برابر 1- باشد پشته خالی است.

- int IsEmpty(Stack *A)

```
{
  return (A->top == -1 ? 1 : 0);
}
```

پیچیدگی اعمال مختلف روی پشته ایجاد شده با آرایه

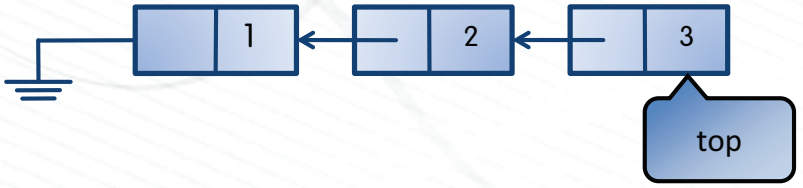
1. ایجاد (Create)، ← O(1)
2. اضافه نمودن یک عضو (Push)، ← O(1)
3. حذف یک عضو (Pop)، ← O(1)
4. دریافت مقدار بالای پشته (Top)، ← O(1)
5. تعیین خالی بودن پشته (IsEmpty) ← O(1)

پیاده سازی با استفاده از فهرست پیوندی

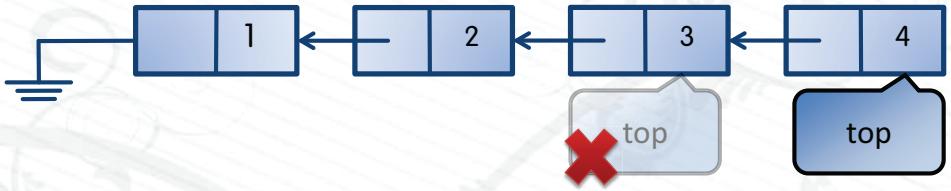
- پیاده سازی پشته با استفاده از فهرست پیوندی
- چگونگی اضافه نمودن به پشته
- نحوه حذف از پشته

پیاده سازی پشته با استفاده از فهرست پیوندی

- برای انجام این کار ما فقط به یک فهرست پیوندی احتیاج داریم و هیچ متغیر اضافی لازم نیست.
- فهرست را چنان پیاده سازی می‌کنیم که
  1. هر گره به گره قبلی اشاره می‌کند،
  2. گره اول به NULL اشاره کند، و
  3. Top همان head فهرست پیوندی باشد.



چگونگی اضافه نمودن به پشته (Push)



- برای اضافه نمودن گره‌ای جدید بعد از گره دوم باید
  1. گره‌ای جدید ساخت، به داده‌های آن مقدار داد،
  2. اشاره‌گر گره جدید به گره top اشاره کند، و
  3. top برابر این گره جدید می‌شود.

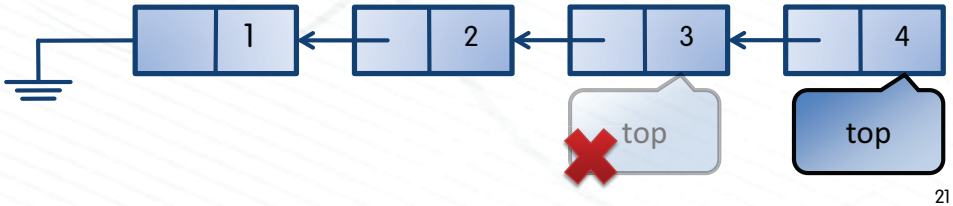
الگوریتم اضافه نمودن به پشته

- void Push(Stack *S, int a)

{

- Node *t = new Node;
- t->Data = a;
- t->Next = S->top;
- S->top=t;

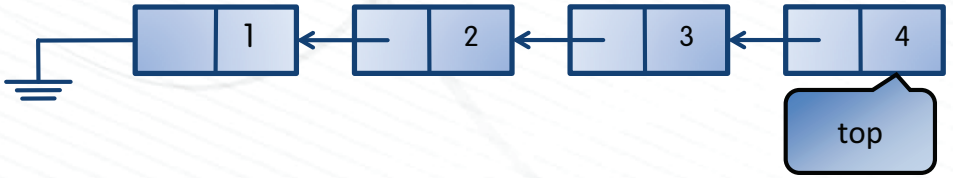
}



نحوه حذف عنصر top (Pop)

- برای حذف عنصر top باید

1. نشانی گره top را ذخیره نمود،
2. گره top را برابر top->Next قرار داد، و
3. نشانی ذخیره شده را از حافظه پاک نمود.



یافتن مقدار top (Top)

- int Top(Stack *A)

{

- return A->top->Data;

}

دقت شود که تابع Top چیزی حذف یا اضافه نمی‌کند!

الگوریتم حذف عنصر Top (Pop)

- void Pop(Stack *S)

{

- Node *t;
- t = S->top;
- S->top = S->top->Next; // or S->top = t->Next;
- free(t);

}

چگونگی تشخیص خالی بودن پشته (IsEmpty)

- در ابتدا مقدار top را برابر NULL قرار می‌دهیم؛ از آنجا که
  - با اضافه نمودن عنصر جدید این متغیر به آن اشاره خواهد نمود و
  - با حذف عنصر این متغیر به ماقبل خود اشاره خواهد نمود
- بنابراین هرگاه این متغیر برابر NULL باشد پشته خالی است.

```
int IsEmpty(Stack *S)
{
    return (S->top == NULL ? 1 : 0);
}
```


فهرست پیوندی – زمان مورد نیاز اعمال مختلف

1. ایجاد (Create)، ← O(1)
2. اضافه نمودن یک عضو (Push)، ← O(1)
3. حذف یک عضو (Pop)، ← O(1)
4. دریافت مقدار بالای پشته (Top)، ← O(1)
5. تعیین خالی بودن پشته (IsEmpty) ← O(1)