

Data Structure

Data Structure

- It is a logical way of organizing data that makes them efficient to use.
- Addition
- Modification
- Deletion
- Searching
- Traversing (display)

STACK AND QUEUE

STACK

- A Stack is linear structure implemented in LIFO(last in first out). A stack is a dynamic data structure, bcoz it an grow(increase number of elements) or shrink(decrease number of elements). A static data structure has fixed size.
- LIFO-means element last inserted would be the first one to be deleted.
- Two rules of Stack:
 - 1. Data can only be removed from top of stack(pop operation)
 - 2. New element can only be added to top of stack(push operation)

12

10

9

2

7

3

4

5

STACK

- Some terms related to Stack:
- **Peek/inspection**-inspecting the value at stack's top without removing it.
- **Overflow**: It is a situation when someone tries to push an item in stack, when stack is fixed and can't grow further or no memory left for new item.
- **Underflow**: It is a situation when someone tries to pop an item in empty stack.

Stack Application

- Stack can be used for various purposes. e.g.
- **Reversing a line-** to push each character on to a stack as it is read. When the line is finished, characters are then popped off the stack.
- **Polish Strings-** convert arithmetic expressions into polish strings by using stacks. It is of two types: postfix and prefix.
- **To reverse a word.**
- **An "undo" mechanism in text editors;** this operation is accomplished by keeping all text changes in a **stack**. ...
- **A stack of plates/books in a cupboard.**
- **Wearing/Removing Bangles.**

Types of Stack

- An item stored in a stack is also called item-code. We can implement stack which contain group information.

- Stack of integers
4
5
6
8

- Stack of strings
'a'
'b'
'c'
'd'

- Stack of lists
1,'abc',98
2,'xyz',78
3,'def',98

Implementing Stack in Python

- We use lists to implement Stack.
- #to insert element into stack
- `s=[]`
- `def push(s,e):`
 - `s.append(e)`
 - `top=len(s)-1`

Implementing Stack in Python

- We use lists to implement Stack.
- #to delete element from stack
- s=[]
- def pop(s):
 - if (s==[]):
 - print("underflow")
 - else:
 - e=s.pop()
 - if(len(s)==0):
 - top=None
 - else:
 - top=len(s)-1
 - return e

Implementing Stack in Python

- We use lists to implement Stack.
- #to display element of stack

```
s=[]
```

```
def display(s):
```

```
    if(s==[]):
```

```
        print("underflow")
```

```
    else:
```

```
        top=len(s)-1
```

```
        print(s[top],"<--top")
```

```
        for d in range(top-1,-1,-1):
```

```
            print(s[d])
```