```
#implement an ADT and computer space and time complexities
import time
start=time.time()
import tracemalloc

def push(stk,itm):
    stk.append(itm)
    print("element to be inserted")
    print(stk)

def pop(stk):
    e=stk.pop()
    print("poping element:",e)

def peek(stk):
    print("the element at peek:",stk[-1])

def display(stk):
    print("stack element:",stk)

def isempty(stk):
    if len(stk)==0:
        print("stack isempty")
    else:
        print("stack contains",len(stk),"elements")
tracemalloc.start()
stack=[10,20,30,40,50,2]
while True:
    print("--------- stack operation---------")
    print("1.push")
    print("2.pop")
    print("3.peek")
    print("4.display")
    print("5.isempty")
    print("6.exit")

    ch=int(input("enter your choice:"))
    if ch==1:
        item=int(input("entre element:"))
        push(stack,item)
    if ch==2:
        pop(stack)
    if ch==3:
        peek(stack)
    if ch==4:
        display(stack)
    if ch==5:
        isempty(stack)
    if ch==6:
     break
print("space complexity=",tracemalloc.get_tracemalloc_memory(),"bytes")
tracemalloc.stop()
print("\new")
end=time.time()
print("time complexity=",end-start)
```

**********output**********

```
------------ stack operation----------------------
1.push
2.pop
3.peek
4.display
5.isempty
6.exit
enter your choice:3
the element at peek: 2
--------- stack operation---------
1.push
2.pop
3.peek
4.display
5.isempty
6.exit
enter your choice:4
stack element: [10, 20, 30, 40, 50, 2]
------------ stack operation----------------------
1.push
2.pop
3.peek
4.display
5.isempty
6.exit
enter  your  choice:5
stack contains 6 elements
------------ stack operation----------------------
1.push
2.pop
3.peek
4.display
5.isempty
6.exit
enter your choice:6
space complexity= 15824 bytes
time complexity= 20.14539647102356
```