jupyter005dsa

July 18, 2024

###

Python code for DSA,18/Jul/24

Name: Praanesh Balakrishnan Nair Roll number: BL.EN.U4AIE23123

Linear Search

[1]:
```
import random

def main():
    arr = []
    for i in range(11):
        arr.append(random.randint(0, 10))
    for i in range(len(arr)-1):
        print(arr[i], end = " ")
    print()
    x = linearSearch(arr, int(input("Enter a number to search for in this␣
 ↪random array: ")))
    if x:
        print(f"found at {x}")
    else:
        print("not found")

def linearSearch(array, value):
    for i in range(len(array)):
        if value == array[i]:
            return i




if __name__ == "__main__":
    main()
```

```
0 10 0 10 4 8 4 0 1 9
found at 4
```

Binary Search

[2]:
```
import random

def main():
    arr = []
    for i in range(11):
        arr.append(random.randint(0, 10))
    for i in range(len(arr)-1):
        print(arr[i], end = " ")
    print()
    if binarySearch(arr, int(input("Enter a number to search for in this random␣
 ↪array: "))):
        print("Found")
    else:
        print("not found")

def binarySearch(array, value):
    array = sorted(array)
    lo = 0
    hi = len(array) - 1
    while lo <= hi:
        mid = int((lo + hi)/2)
        if value == array[mid]:
            return True
        elif value < array[mid]:
            hi = mid - 1
        else:
            lo = mid + 1
    return False





if __name__ == "__main__":
    main()
```

```
9 2 5 9 9 9 6 3 9 0
Found
```

Stack Operations

[3]:
```
def main():
    stk = Stack()
    for i in range(5):
        stk.push(i * 10)

    stk.printStack()
```

```python
class Stack:
    def __init__(self):
        self.stack = []
        self.TOP = -1

    def push(self, value):
        self.TOP += 1
        if len(self.stack) > self.TOP:
            self.stack[self.TOP] = value
        else:
            self.stack.append(value)

    def pop(self):
        if self.TOP == -1:
            return None
        value = self.stack[self.TOP]
        self.TOP -= 1
        return value

    def peek(self):
        if self.TOP == -1:
            return None
        return self.stack[self.TOP]

    def printStack(self):
        for i in range(self.TOP, -1, -1):
            print(self.stack[i])



if __name__ == "__main__":
    main()
```

```
40
30
20
10
0
```