

프로그래밍 언어 HW1

B811181 조예린

March 30, 2021

1 과제 1

리스트를 입력 받아 Binary Search로 숫자 찾는 문제

1.1 리스트 입력 받기

map(), split() 이라는 함수를 통해 input()으로 입력 받고 int형으로 변환해주었다.

```
my_list = list(map(int, input().split()))
```

1.2 BS 함수

리스트와 찾을 숫자를 인자로 받는 BS 함수를 정의하였다.

```
def BS(list, key):
    low=0
    high=len(list)-1
    # Search key!
    while low<=high:
        m = int((low + high) / 2)
        if list[m]==key:
            break
        elif list[m]<key:
            low=m+1
        elif list[m]>key:
            high=m-1

    if low>high:
        print("None")
    else:
        print(m+1)
```

2 과제 2

리스트를 Quick Sort로 정렬

2.1 리스트 입력 받기

1.1과 동일

2.2 QuickSort 함수

Recursion으로 QuickSort 함수 구현 인자로는 리스트의 처음과 끝 index를 받았다.

```
# Quick Sort
def QuickSort(first_i , last_i):
    if(first_i<last_i):
        # Partition
        pivot=list [ first_i ]
        i=first_i+1
        j=last_i

        while i<=j:
            while i<=last_i and list [i]<pivot:
                i+=1
            while j>first_i and list [j]>pivot:
                j-=1

            if i>j:
                # Exchange pivot & list [j]
                list [ first_i]=list [ j ]
                list [ j]=pivot
            else:
                # Exchange list [i] & list [j]
                tmp=list [ i ]
                list [ i]=list [ j ]
                list [ j]=tmp

        # Recursion
        QuickSort ( first_i , j-1)
        QuickSort (j+1, last_i)
```

3 과제 3

리스트를 Merge Sort로 정렬

3.1 리스트 입력 받기

1.1과 동일

3.2 MergeSort 함수

Recursion으로 MergeSort 함수 구현 인자로 리스트를 받고 리스트를 반환하였다.

```
#Merge Sort
def MergeSort( list ):
    if len( list )<=1:
        return list
    else :
        m_i=int( len( list )/2)

        # Recursion
        pre=MergeSort( list [: m_i] )
        post=MergeSort( list [m_i:] )

        return Merge( pre , post )
```

3.3 Merge 함수

두 리스트를 받아 Merge한 후 하나의 리스트로 반환하였다.

```
# Merge
def Merge( pre , post ):
    t_list = []
    i=0
    j=0

    while i<len(pre) and j<len(post):
        if pre[i]<=post[j]:
            t_list.append(pre[i])
            i+=1
        else :
            t_list.append(post[j])
            j+=1

    if i>=len(pre):
        t_list.extend(post[j:])
    elif j>=len(post):
        t_list.extend(pre[i:])

    return t_list
```

4 과제 4

이진 트리를 생성하고 각 순회 결과 출력하기

4.1 클래스로 트리 구현

트리의 노드를 클래스로 구현하여 각 노드를 리스트에 추가 하였다. Node 클래스의 멤버변수는 key (key값), L(왼쪽 자식 노드), R(오른쪽 자식 노드)이 있다.

```
class Node:
    def __init__(self, Key, L, R):
        self.key=Key
        self.L=L
        self.R=R
```

4.2 순회하기

각 순회 함수를 Recursion으로 구현하여 결과를 출력하였다. PreOrder 함수를 예시로 첨부

```
def PreOrder(now):
    print(my_tree[now].key)
    if my_tree[now].L!=-1:
        PreOrder(my_tree[now].L)
    if my_tree[now].R!=-1:
        PreOrder(my_tree[now].R)
```

5 과제 5

강의실 배정하기

5.1 이중리스트로 입력 받기

각 강의의 번호와 시작, 종료 시간을 리스트로 입력 받은 후 다시 classes라는 리스트에 추가하였다.

```
# input
n=int(input())
classes=[]
for i in range(0, n, 1):
    my_class = list(map(int, input().split()))
    classes.append(my_class)
```

5.2 시작 시간을 기준으로 정렬

itemgetter를 통해 classes를 각 강의의 시작 시간을 기준으로 정렬하였다.

```
from operator import itemgetter
classes.sort(key=itemgetter(1))
```

5.3 강의실 배정

nowtime을 현재 들여다보는 강의의 시작으로 갱신해가며 강의 배정을 한다. playing에 있는 강의가 끝나면 result에 추가해준다.

```
now_time=0
playing=classes[0]
result=[]

for cl in classes:
    now_time=cl[1]

    # playing class is finished,
    if playing[2]<=now_time:
        result.append(playing[0])
        playing=cl
    # change playing class.
    elif playing[2]>cl[2]:
        playing=cl
    # append last class.
result.append(playing[0])
```

6 과제 6

6.1 입력 받기

str 리스트로 입력 받았다.

6.2 암호화 검사

test 함수에 검사할 code를 넣어준다. 암호화 되지 않은 경우 code를 '-'로 바꿔 주고, 암호화 된 경우 len(code)는 0이다.

```
def code_test(code):
    # 100
    if code[:3]=='100':
        code=code[3:]
    else:
        code='-'
    return code

    # ~
    while code[0]=='0':
        code=code[1:]

    # 1
    if code[0]=='1':
```

```

        code = code[1:]
else:
    code='-'
    return code

# ~|01
# ~1
if code[0]== '1':
    while code[0]== '1':
        code = code[1:]
    if code[0]== '0':
        code.insert(0, '1')

# 01
elif code[0]== '0':
    if code[:2]== '01':
        code = code[2:]
    else:
        code='-'
        return code
else:
    code='-'
    return code

return code

```