

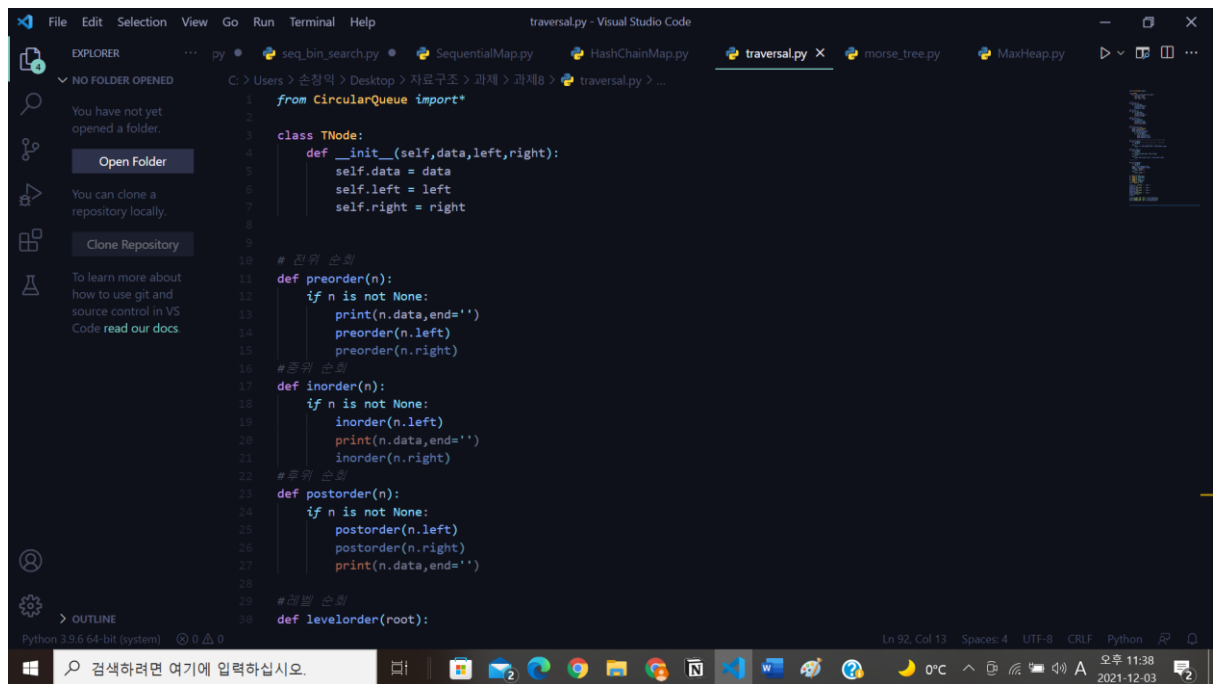
자료구조 8장 과제

바이오메디컬공학부

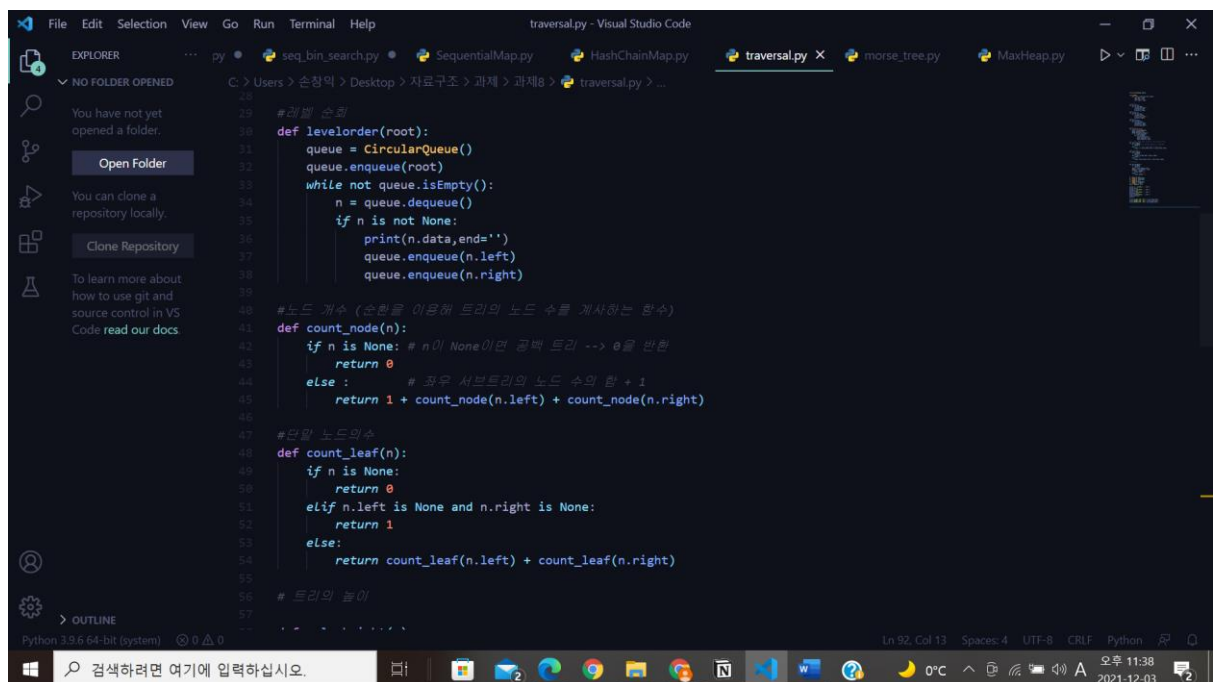
201804594 손무현

VSCode로 과제를 진행하였습니다.

1. 순회



```
1 from CircularQueue import*
2
3 class TNode:
4     def __init__(self,data,left,right):
5         self.data = data
6         self.left = left
7         self.right = right
8
9
10 # 전위 순회
11 def preorder(n):
12     if n is not None:
13         print(n.data,end='')
14         preorder(n.left)
15         preorder(n.right)
16
17 # 중위 순회
18 def inorder(n):
19     if n is not None:
20         inorder(n.left)
21         print(n.data,end='')
22         inorder(n.right)
23
24 # 후위 순회
25 def postorder(n):
26     if n is not None:
27         postorder(n.left)
28         postorder(n.right)
29         print(n.data,end='')
30
31 # 레벨 순회
32 def levelorder(root):
```



```
33     queue = CircularQueue()
34     queue.enqueue(root)
35     while not queue.isEmpty():
36         n = queue.dequeue()
37         if n is not None:
38             print(n.data,end='')
39             queue.enqueue(n.left)
40             queue.enqueue(n.right)
41
42 #노드 개수 (순회를 이용해 트리의 노드 수를 계산하는 함수)
43 def count_node(n):
44     if n is None: # n이 None이면 공백 트리 --> 0을 반환
45         return 0
46     else: # 좌우 서브트리의 노드 수의 합 + 1
47         return 1 + count_node(n.left) + count_node(n.right)
48
49 #단말 노드의 수
50 def count_leaf(n):
51     if n is None:
52         return 0
53     elif n.left is None and n.right is None:
54         return 1
55     else:
56         return count_leaf(n.left) + count_leaf(n.right)
57
58 # 트리의 높이
```

```

55 # 트리의 높이
56 def calc_height(n):
57     if n is None:
58         return 0
59     hLeft = calc_height(n.left)
60     hRight = calc_height(n.right)
61     if(hLeft > hRight):
62         return hLeft + 1
63     else:
64         return hRight + 1
65 # 테스트 프로그램
66 d = TNode('D',None,None)
67 e = TNode('E',None,None)
68 b = TNode('B',d,e)
69 f = TNode('F',None,None)
70 c = TNode('C',f,None)
71 root = TNode('A',b,c)
72 print('\n In-Order : ', end='')
73 inorder(root)
74 print('\n Pre-Order : ', end='')
75 preorder(root)
76 print('\n Post-Order : ', end='')
77 postorder(root)
78 print('\nLevel-Order : ', end='')
79 levelorder(root)
80 print()
81 print("노드의 개수 = %d개" % count_node(root))
82 print("단말의 개수 = %d개" % count_leaf(root))
83 print("트리의 높이 = %d" % calc_height(root))

```

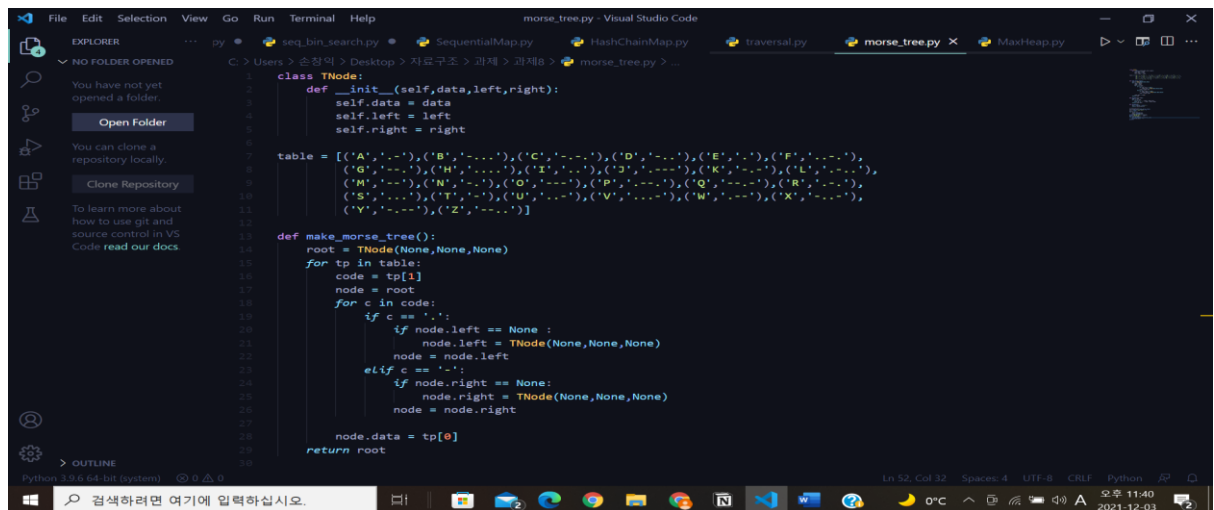
```

PS C:\Users\손창익>
PS C:\Users\손창익>
PS C:\Users\손창익>
PS C:\Users\손창익>
PS C:\Users\손창익>
PS C:\Users\손창익>
PS C:\Users\손창익> & C:/Users/손창익/AppData/Local/Programs/Python/Python39/python.exe c:/Users/손창익/Desktop/자료구조/과제/과제8/traversal.py
In-Order : DBEAFC
Pre-Order : ABDECF
Post-Order : DEBFCA
Level-Order : ABCDEF
노드의 개수 = 6개
단말의 개수 = 3개
트리의 높이 = 3
PS C:\Users\손창익>

```

시스템에서 만들어 놓은 Queue를 사용하지 않고 5장 과제에서 구현했던 CircularQueue 파일을 복사하여 해당 프로그램과 같은 폴더에 위치하도록 하여 import하는 방식으로 코드를 작성했습니다.

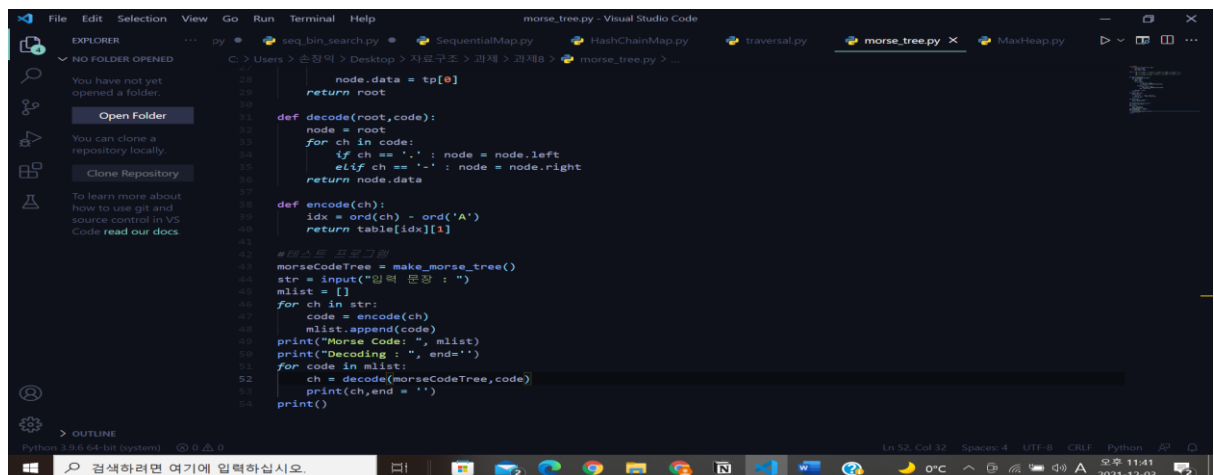
2. 모스 부호 구현 및 실행



```
File Edit Selection View Go Run Terminal Help
morse_tree.py - Visual Studio Code

EXPLORER
C:\Users\> Users\> Desktop> 자료구조 > 과제 > 과제8 > morse_tree.py > ...
You have not yet opened a folder.
Open Folder
You can clone a repository locally.
Clone Repository
To learn more about how to use git and source control in VS Code read our docs.

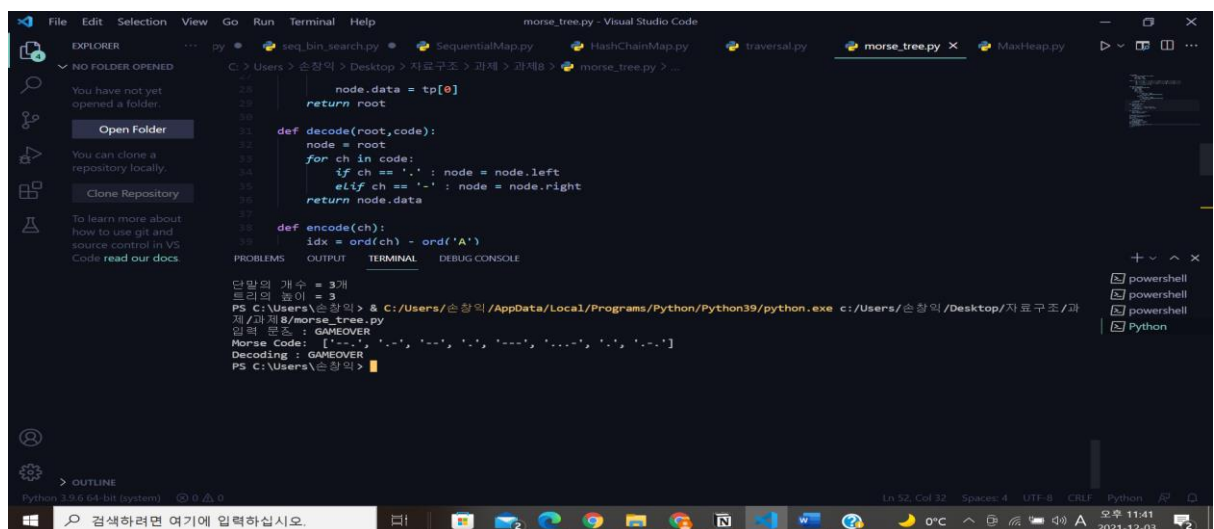
morse_tree.py
1 class TNode:
2     def __init__(self, data, left, right):
3         self.data = data
4         self.left = left
5         self.right = right
6
7 table = [
8     ('A', '-.-'), ('B', '-...'), ('C', '-.-.-'), ('D', '-..-'), ('E', '-. '), ('F', '-.-. '),
9     ('G', '--.-'), ('H', '.... '), ('I', '.-.-'), ('J', '--.-.-'), ('K', '-.-.-'), ('L', '-..-'),
10    ('M', '-.-.-'), ('N', '-.-'), ('O', '---'), ('P', '-.-.-'), ('Q', '-.-.-'), ('R', '-.-.-'),
11    ('S', '...-'), ('T', '-.-'), ('U', '...-'), ('V', '...-'), ('W', '-.-.-'), ('X', '-.-.-'),
12    ('Y', '-.-.-'), ('Z', '-.-.-')
13
14 def make_morse_tree():
15     root = TNode(None, None, None)
16     for tp in table:
17         code = tp[1]
18         node = root
19         for c in code:
20             if c == '.':
21                 if node.left == None:
22                     node.left = TNode(None, None, None)
23                 node = node.left
24             elif c == '-':
25                 if node.right == None:
26                     node.right = TNode(None, None, None)
27                 node = node.right
28         node.data = tp[0]
29     return root
```



```
File Edit Selection View Go Run Terminal Help
morse_tree.py - Visual Studio Code

EXPLORER
C:\Users\> Users\> Desktop> 자료구조 > 과제 > 과제8 > morse_tree.py > ...
You have not yet opened a folder.
Open Folder
You can clone a repository locally.
Clone Repository
To learn more about how to use git and source control in VS Code read our docs.

morse_tree.py
28         node.data = tp[0]
29         return root
30
31 def decode(root, code):
32     node = root
33     for ch in code:
34         if ch == '.': node = node.left
35         elif ch == '-': node = node.right
36     return node.data
37
38 def encode(ch):
39     idx = ord(ch) - ord('A')
40     return table[idx][1]
41
42 # 테스트 프로그램
43 morseCodeTree = make_morse_tree()
44 str = input("입력 문장 : ")
45 mlist = []
46 for ch in str:
47     code = encode(ch)
48     mlist.append(code)
49 print("Morse Code : ", mlist)
50 print("Decoding : ", end='')
51 for code in mlist:
52     ch = decode(morseCodeTree, code)
53     print(ch, end=' ')
54 print()
```



```
File Edit Selection View Go Run Terminal Help
morse_tree.py - Visual Studio Code

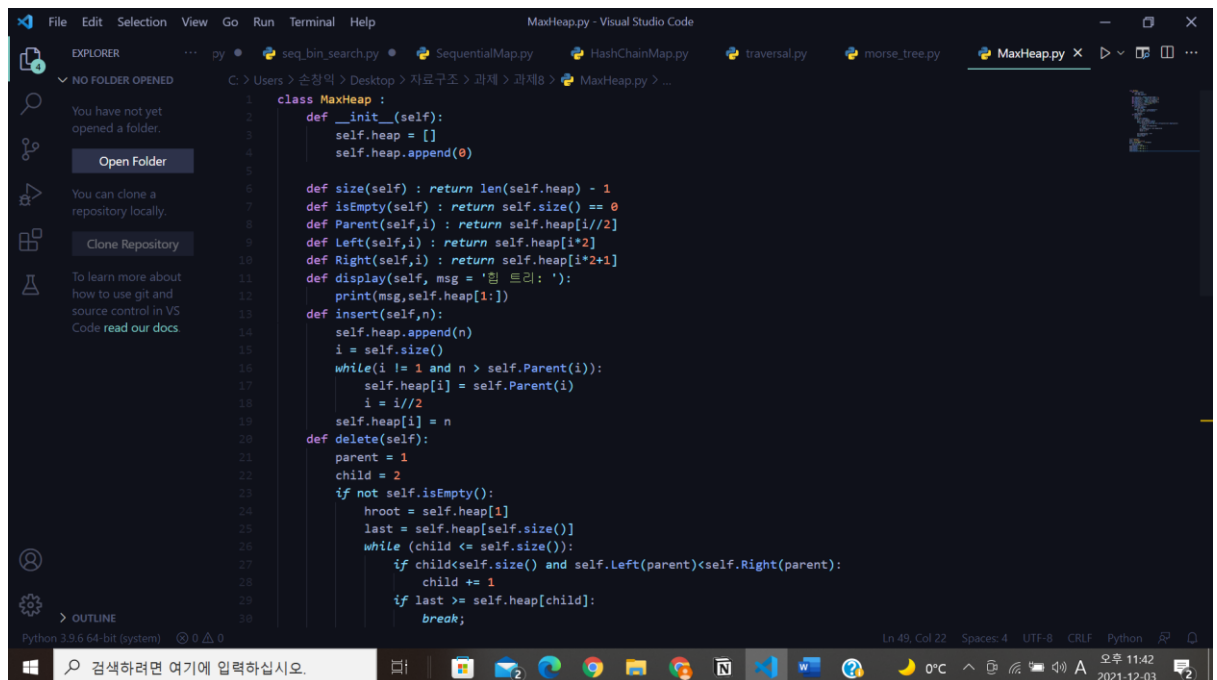
EXPLORER
C:\Users\> Users\> Desktop> 자료구조 > 과제 > 과제8 > morse_tree.py > ...
You have not yet opened a folder.
Open Folder
You can clone a repository locally.
Clone Repository
To learn more about how to use git and source control in VS Code read our docs.

morse_tree.py
28         node.data = tp[0]
29         return root
30
31 def decode(root, code):
32     node = root
33     for ch in code:
34         if ch == '.': node = node.left
35         elif ch == '-': node = node.right
36     return node.data
37
38 def encode(ch):
39     idx = ord(ch) - ord('A')
40     return table[idx][1]
41
42 # 테스트 프로그램
43 morseCodeTree = make_morse_tree()
44 str = input("입력 문장 : ")
45 mlist = []
46 for ch in str:
47     code = encode(ch)
48     mlist.append(code)
49 print("Morse Code : ", mlist)
50 print("Decoding : ", end='')
51 for code in mlist:
52     ch = decode(morseCodeTree, code)
53     print(ch, end=' ')
54 print()

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
단말의 개수 = 3개
트리의 높이 = 3
PS C:\Users\> & C:\Users\> AppData\Local\Programs\Python\Python39\python.exe c:/Users/> Desktop/> 자료구조/과제8/morse_tree.py
입력 문장 : GAMEOVER
Morse Code : ['-.-.', '-.-.', '-.-.', '-.-.-', '-.-.-', '-.-.-', '-.-.-']
Decoding : GAMEOVER
PS C:\Users\>
```

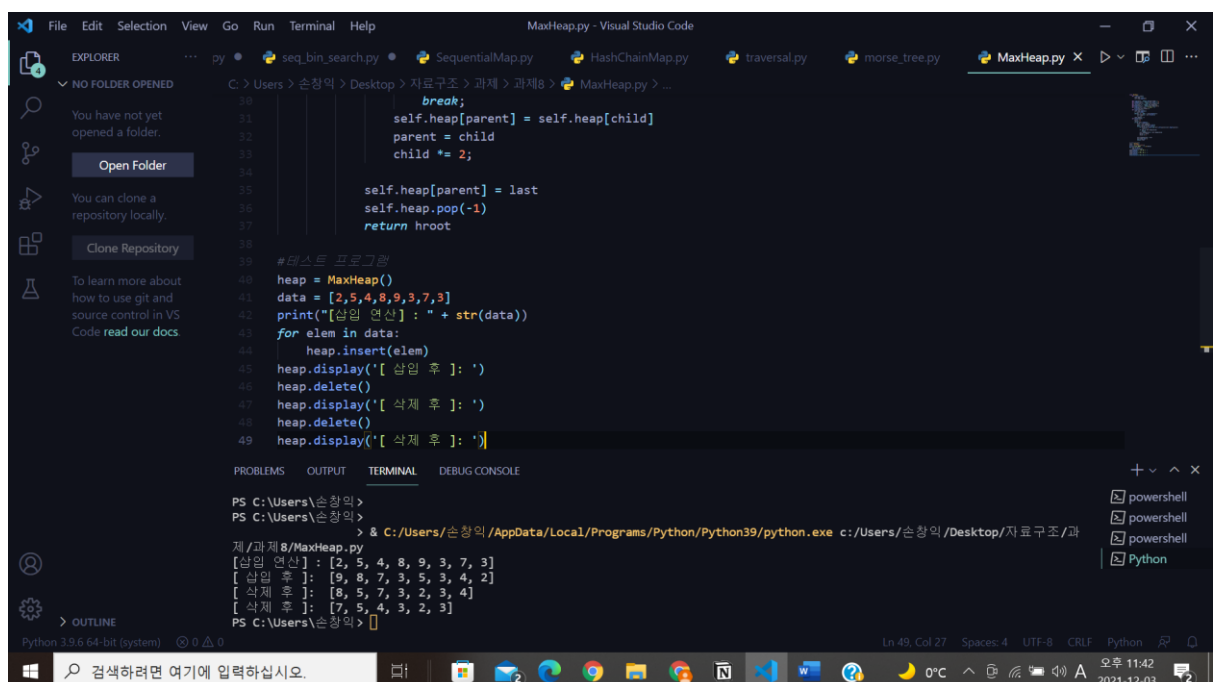
모스부 부호를 모스 코드의 결정 트리 알고리즘으로 구현하고 테스트 프로그램을 작성하여 정상적으로 실행이 되는 것을 확인할 수 있었습니다.

3. 힙 구현 및 실행



```
class MaxHeap :
    def __init__(self):
        self.heap = []
        self.heap.append(0)

    def size(self) : return len(self.heap) - 1
    def isEmpty(self) : return self.size() == 0
    def Parent(self,i) : return self.heap[i//2]
    def Left(self,i) : return self.heap[i*2]
    def Right(self,i) : return self.heap[i*2+1]
    def display(self, msg = '힙 트리: '):
        print(msg,self.heap[1:])
    def insert(self,n):
        self.heap.append(n)
        i = self.size()
        while(i != 1 and n > self.Parent(i)):
            self.heap[i] = self.Parent(i)
            i = i//2
        self.heap[i] = n
    def delete(self):
        parent = 1
        child = 2
        if not self.isEmpty():
            hroot = self.heap[1]
            last = self.heap[self.size()]
            while (child <= self.size()):
                if child<self.size() and self.Left(parent)<self.Right(parent):
                    child += 1
                if last >= self.heap[child]:
                    break;
```



```
break;
self.heap[parent] = self.heap[child]
parent = child
child *= 2;

self.heap[parent] = last
self.heap.pop(-1)
return hroot

# 테스트 프로그램
heap = MaxHeap()
data = [2,5,4,8,9,3,7,3]
print("[삽입 연산] : " + str(data))
for elem in data:
    heap.insert(elem)
heap.display('[ 삽입 후 ]: ')
heap.delete()
heap.display('[ 삭제 후 ]: ')
heap.delete()
heap.display('[ 삭제 후 ]: ')
```

```
PS C:\Users\손창익>
PS C:\Users\손창익> & C:/Users/손창익/AppData/Local/Programs/Python/Python39/python.exe c:/Users/손창익/Desktop/자료구조/과제8/MaxHeap.py
[삽입 연산] : [2, 5, 4, 8, 9, 3, 7, 3]
[ 삽입 후 ]: [9, 8, 7, 3, 5, 3, 4, 2]
[ 삭제 후 ]: [8, 5, 7, 3, 2, 3, 4]
[ 삭제 후 ]: [7, 5, 4, 3, 2, 3]
PS C:\Users\손창익>
```

최대 힙을 클래스로 구현하는 코드를 작성하였고 마찬가지로 테스트 프로그램을 작성하여 실행시켰을 때 교재 내용과 같은 결과가 출력되는 것을 확인할 수 있었습니다.

이번 과제를 통해 순회를 구현하고 이진트리의 응용으로 모르스 코드를 구현해봤으며 힙 트리에서 최대 힙을 직접 구현해보면서 수업시간에 배운 내용에 조금 더 익숙해질 수 있었습니다. 하지만 다른 단원보다 내용이 어렵게 느껴져 계속해서 연습하고 코드를 계속해서 직접 구현 해야겠다는 것을 느꼈습니다.