

Binary Search Tree 구현

파일처리론 01분반
컴퓨터공학과 2142978 채종화

목차

1. 구현한 프로그램의 설명
2. 실행 결과
3. 구현에 있어서 어려웠던 점과 해결 방법

프로그램 설명

1. 프로그램의 개요: bst.py를 실행하면 같은 디렉토리 내의 bst_input.txt에서 입력받은 값들을 읽어 binary search tree의 규칙에 따라 검색, 삽입, 삭제 값을 bst_output.txt에 출력

2. 코드 구성

2.1 Node 클래스

- 노드의 값, 왼쪽 자식, 오른쪽 자식 저장

2.2 BinarySearchTree클래스

- insert(value): 트리에 새로운 값 삽입, 값을 비교하여 왼쪽, 오른쪽 트리로 이동하여 삽입 위치를 찾음

- search(value): 트리에서 값을 검색, 검색 경로 반환

- delete(current, value): 트리에서 특정 값을 삭제

2.3 input_file함수

- 입력 파일의 값을 읽음

2.4 output_file함수

- 검색, 삽입, 삭제 결과를 출력

2.5main함수

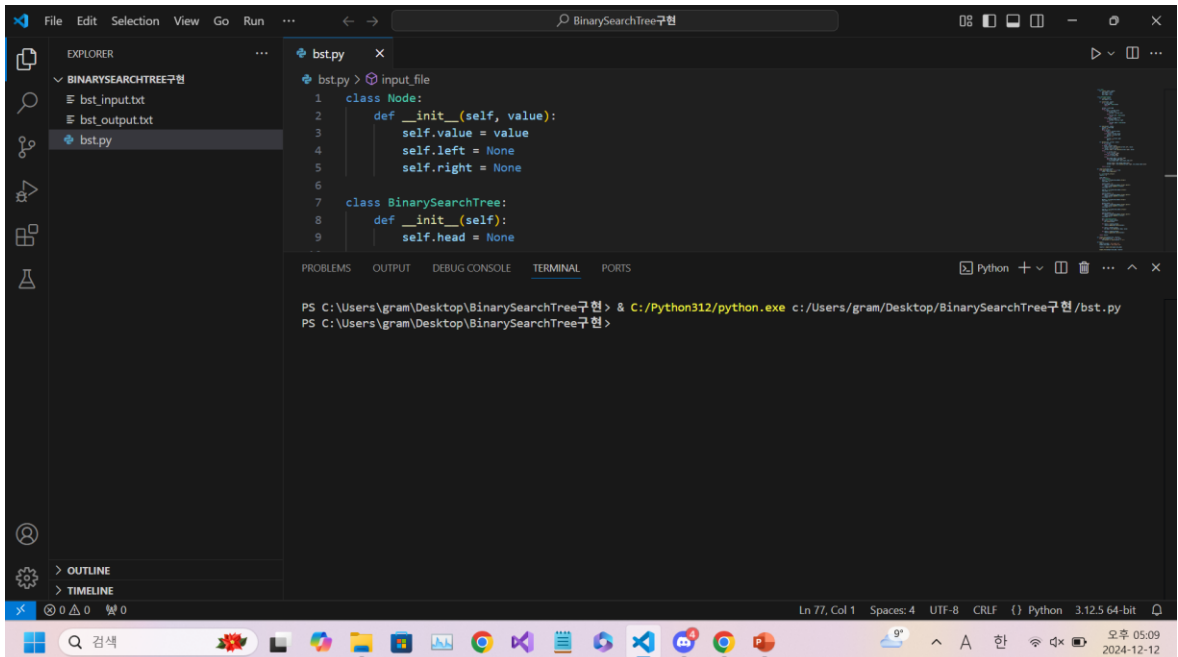
- 입력 파일(bst_input.txt)을 읽음

- binary search tree작업 수행

- 결과를 출력 파일(bst_output.txt)에 저장

3. 사용한 python version: 3.12.5

실행 결과

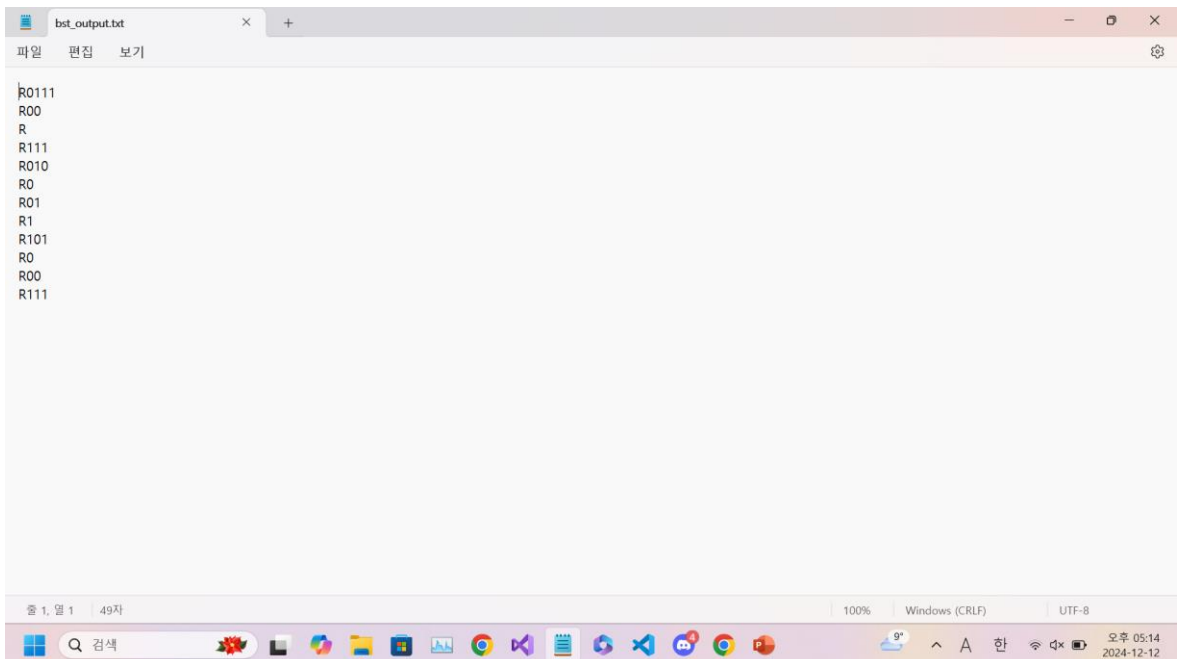


The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left displays a project named 'BINARYSEARCHTREE구현' with files 'bst_input.txt', 'bst_output.txt', and 'bst.py'. The main editor window shows the code in 'bst.py', which defines a 'Node' class and a 'BinarySearchTree' class. The terminal window at the bottom shows the command to run the script: 'PS C:\Users\gram\Desktop\BinarySearchTree구현> & C:/Python312/python.exe c:/Users/gram/Desktop/BinarySearchTree구현/bst.py'. The status bar at the bottom indicates the file is at line 77, column 1, using UTF-8 encoding and CRLF line endings.

```
1 class Node:
2     def __init__(self, value):
3         self.value = value
4         self.left = None
5         self.right = None
6
7 class BinarySearchTree:
8     def __init__(self):
9         self.head = None
```

```
PS C:\Users\gram\Desktop\BinarySearchTree구현> & C:/Python312/python.exe c:/Users/gram/Desktop/BinarySearchTree구현/bst.py
PS C:\Users\gram\Desktop\BinarySearchTree구현>
```

터미널 결과(vscord)



The screenshot shows a text editor window titled 'bst_output.txt'. The file contains the output of the program, which is a list of nodes in a binary search tree. The output is as follows:

```
R0111
R00
R
R111
R010
R0
R01
R1
R101
R0
R00
R111
```

The status bar at the bottom indicates the file is at line 1, column 1, using Windows (CRLF) line endings and UTF-8 encoding.

생성된 bst_output.txt

구현에 있어서 어려웠던 점과 해결 방법

-어려웠던 점 1: 트리를 어떻게 구현할 것인지에 대해 어려움을 겪음

-해결 방법: 리스트로 트리 구성을 시도하였지만 삭제 구현에 실패, 각 키의 왼쪽 오른쪽 자식을 참조하도록 구현-각 키 간의 참조를 통해 간접적으로 트리 구조를 구현

-어려웠던 점 2: 삭제 구현에 있어 삽입이나 검색보다 훨씬 더 어려움을 겪음

-해결 방법: if문을 이용하여 자식이 없는 경우, None을 반환하고, 자식이 왼쪽 또는 오른쪽에 하나만 있는 경우에는 자식을 반환, 두 자식이 있는 경우에는 오른쪽 서브트리의 최솟값을 반환하도록 함

이상입니다