

Data Structure

Project #1



제출일 :2016.10.7

팀장 :김병휘(2013722029)

팀원 :김정환(2014722102)

조원희(2011603051)

1.Introduction

이번 프로젝트 이진 탐색 트리와 환형 연결 리스트, Queue 세 개의 자료 구조를 가지고 단어장을 관리하는 프로그램이다. 단어장에는 총 3가지가 있으며 외워야 할 단어장(TO_MEMORIZE)은 Queue를 이용하여 구축하고 외우고 있는 단어장(MEMORIZING)은 이진 탐색 트리로 구축하고 마지막 이전에 외운 단어장(MEMORIZED)은 환형 연결 리스트로 구축한다.

[1] 사용되는 자료 구조

1) Queue

선입 선출의 자료구조로 영어 단어와 뜻을 들어오는 순서대로 저장하고 있다.

2) BST(Binary Search Tree)

i) 알파벳 BST

하나의 노드당 최대 2개의 자식을 가질 수 있는 자료구조로 Key값으로 알파벳을 가지고 있으며 data로 단어 BST를 가지고 있다.

ii) 단어 BST

알파벳 BST와 마찬가지로 최대 2개의 자식을 가질 수 있으며 Key값으로는 영어단어를 가지고 있으며 data로 뜻을 가지고 있다.

3) 환형 연결 리스트(Circular Linked List)

기본적으로는 리스트의 구조를 가지고 있으며 마지막 노드는 다음 노드의 위치를 header의 위치를 가리킨다. 환형 연결 리스트는 영어 단어와 뜻을 data로 가지고 있다.

[2] 명령어

1) LOAD

기존의 단어장 정보를 불러오는 명령어로, to_memorize_word.txt, memorizing_word.txt, memorized_word.txt에 들어 있는 단어 정보를 각각 Queue, BST, Circular Linked List에 이전과 동일한 연결 순서를 가지도록 저장한다.

만약 텍스트 파일이 존재하지 않거나 자료구조에 이미 데이터가 있는 경우에는 에러 코드를 출력한다.

2) ADD

단어 텍스트 파일에 있는 단어 정보를 읽어오는 명령어로, word.txt의 단어 정보를 모두 TO_MEMORIZE(Queue)에 저장한다.

만약 파일이 존재하지 않거나 파일에 단어가 존재하지 않는다면 에러코드를 출력한다.

3) MOVE 숫자

사용자가 입력한 수 만큼 TO_MEMORIZE의 단어 정보를 MEMORIZING으로 옮기는 명령어로, 입력하는 수는 1~100사이 정수이고 입력 받은 수와 MEMORIZING의 단어의 수의 합은 100을 넘지 않도록 한다. 만약 사용자가 입력한 수만큼 TO_MEMORIZE에 단어가 존재하지 않는 경우나 입력 받은 수와 MEMORIZING의 수의 합이 100을 넘는 경우에 에러 코드를 출력한다.

4) SAVE

현재 단어장 정보를 저장하는 명령어로, Queue, BST, Circular Linked List의 정보를 to_memorize_word.txt, memorizing_word.txt, memorized_word.txt에 저장한다. BST의 경우에는 알파벳 BST를 기준으로 pre-order로 순회하며 저장한다. 만약 단어장 정보가 존재하지 않을 경우에는 에러 코드를 출력한다.

5) TEST 단어 뜻

단어를 외웠는지 테스트하는 명령어로, MEMORIZING에 있는 단어와 뜻이 맞는지 확인하고 맞을 경우 그 단어를 MEMORIZED로 옮긴다. 만약 입력한 단어가 존재하지 않거나 뜻이 틀린 경우에는 에러 코드를 출력한다.

6) SEARCH

단어의 뜻을 찾아 출력하는 명령어로, Queue, BST, Circular Linked List에 입력한 단어가 존재할 경우 해당 단어와 뜻을 출력한다. 만약 입력한 단어가 존재하지 않을 경우에는 에러 코드를 출력한다.

7) PRINT 단어장종류 (형식)

입력한 단어장에 있는 단어들을 출력하는 명령어로, TO_MEMORIZE나 MEMORIZED의 경우에는 들어 있는 단어 정보를 순서대로 전부 출력한다. MEMORIZING의 경우에는 추가로 출력형식이 정해지는데 다음과 같다.

R_PRE: Recursive pre-order(재귀함수를 이용한 pre-order)

I_PRE: Iterative pre-order(반복문을 이용한 pre-order, 재귀함수 사용금지)

R_IN: Recursive in-order(재귀함수를 이용한 in-order)

I_IN: Iterative in-order(반복문을 이용한 in-order, 재귀함수 사용금지)

R_POST: Recursive post-order(재귀함수를 이용한 post-order)

I_POST: Iterative post-order(반복문을 이용한 post-order, 재귀함수 사용금지)

I_LEVEL: Iterative level-order(반복문을 이용한 level-order)

만약 입력한 단어장 정보가 존재하지 않을 경우에는 에러 코드를 출력한다.

8) UPDATE 단어 뜻

단어의 뜻을 변경하는 명령어로, Queue, BST, Circular Linked List에 입력한 단어가 존재할 경우 해당 단어의 뜻을 입력 받은 뜻으로 바꾸고 결과를 출력한다. 약 단어가 존재하지 않거나 단어장 정보가 존재하지 않을 경우에는 에러 코드를 출력한다.

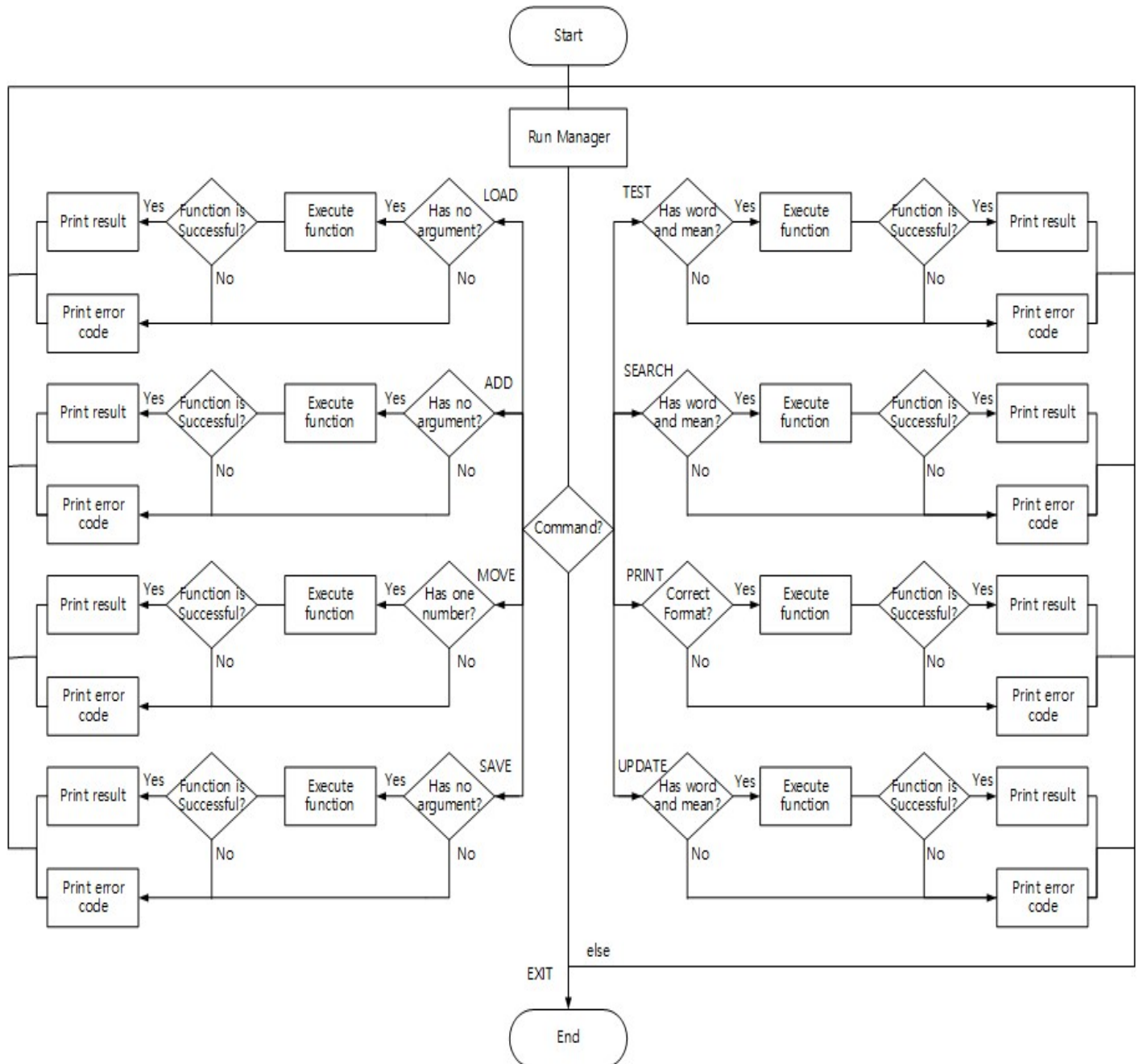
9) EXIT

프로그램 상의 메모리를 해제하며, 프로그램을 종료한다.

[3] 에러코드

명령어	에러 코드
LOAD	100
ADD	200
MOVE	300
SAVE	400
TEST	500
SEARCH	600
PRINT	700
UPDATE	800

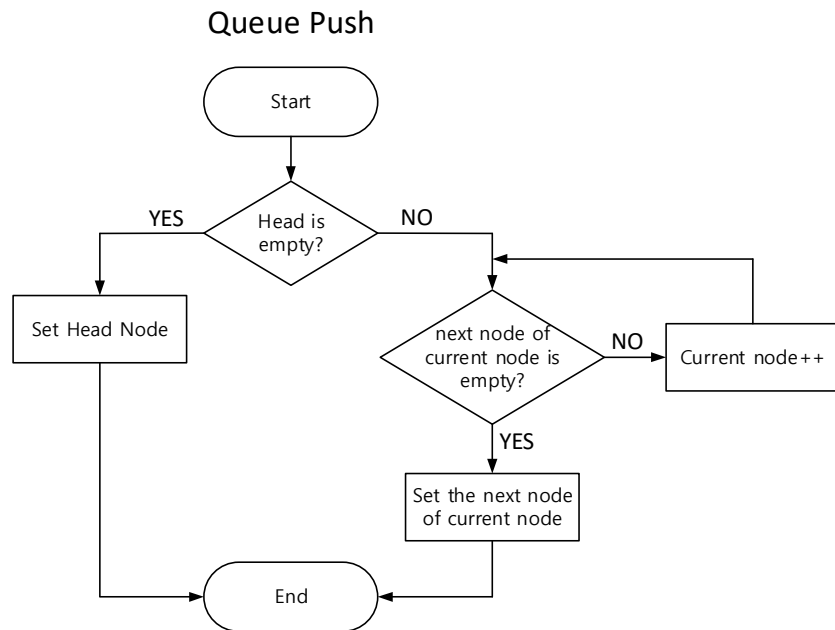
2. Flowchart



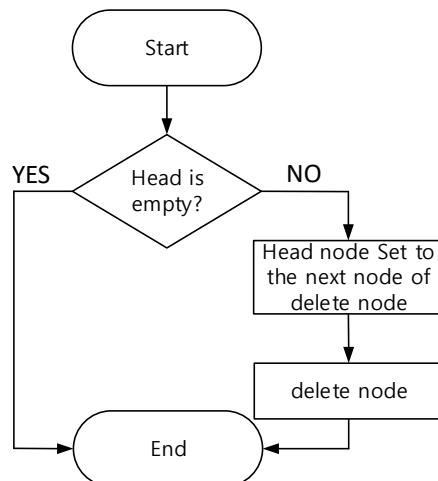
3. Algorithm

1) Queue

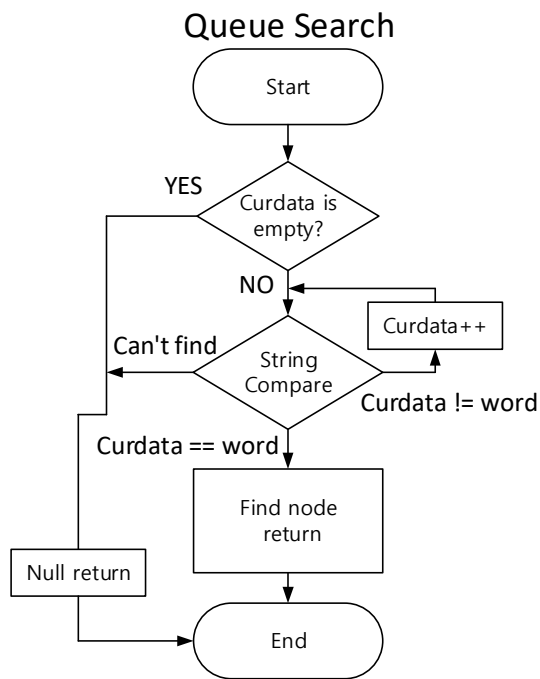
i. Push



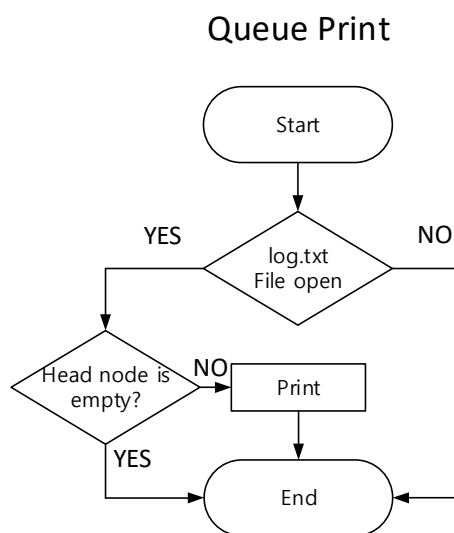
ii. Pop



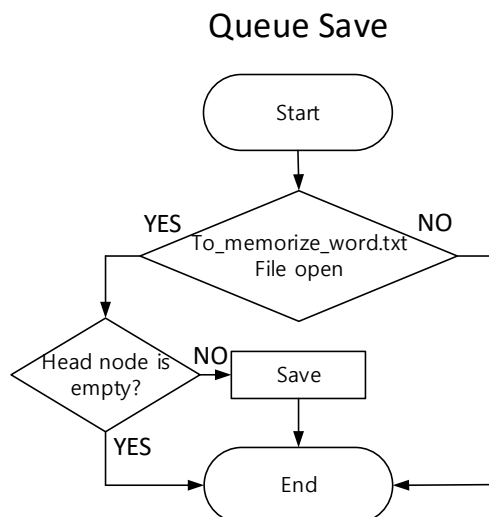
iii. Search



iv. Print

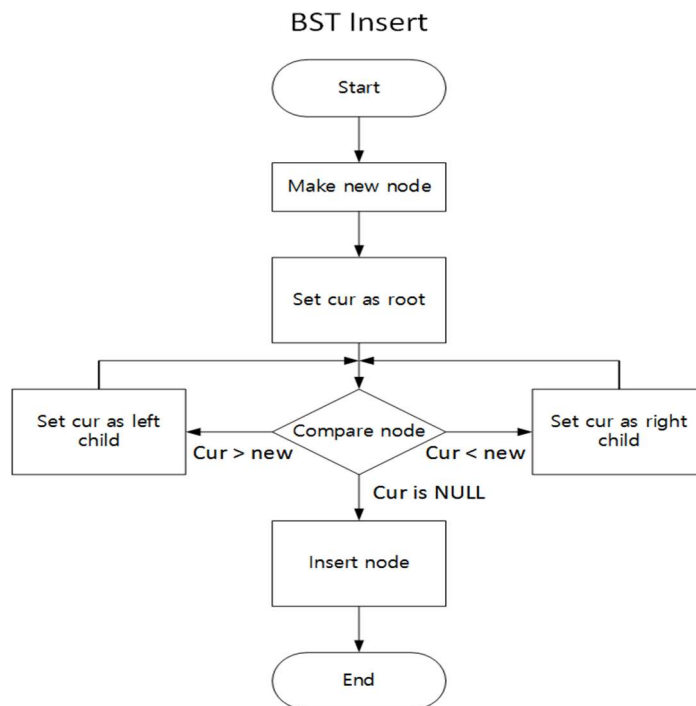


v. Save

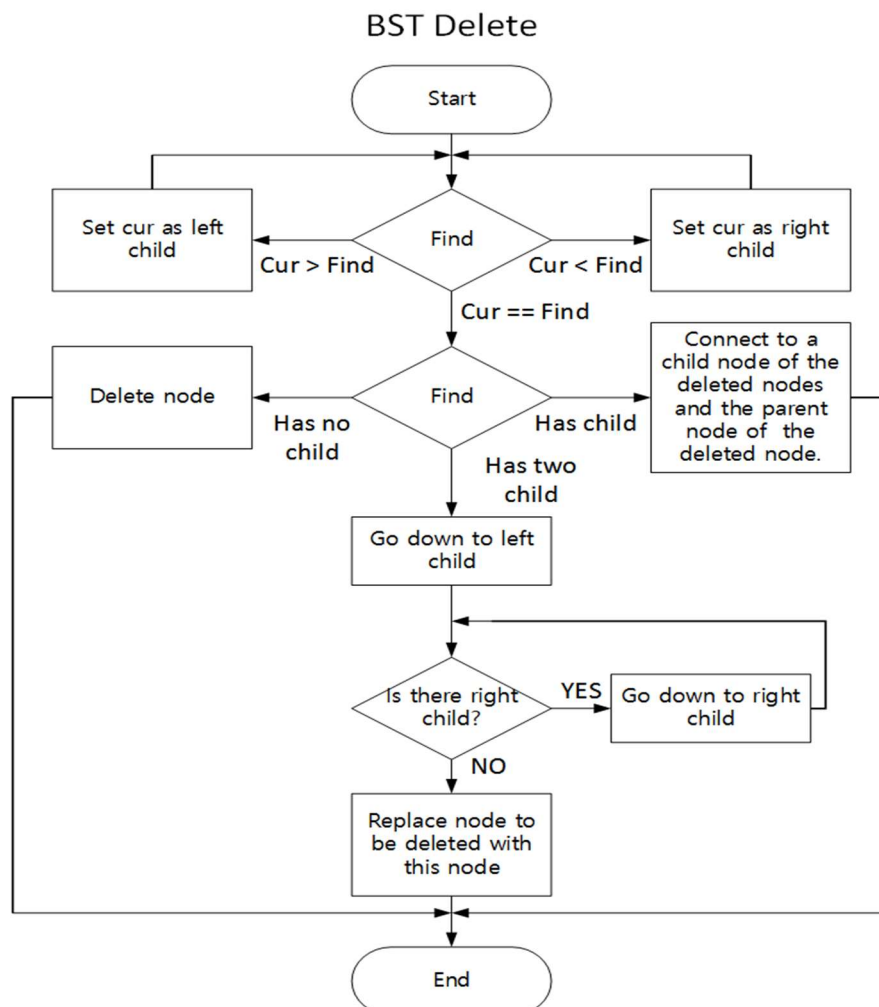


2) BST

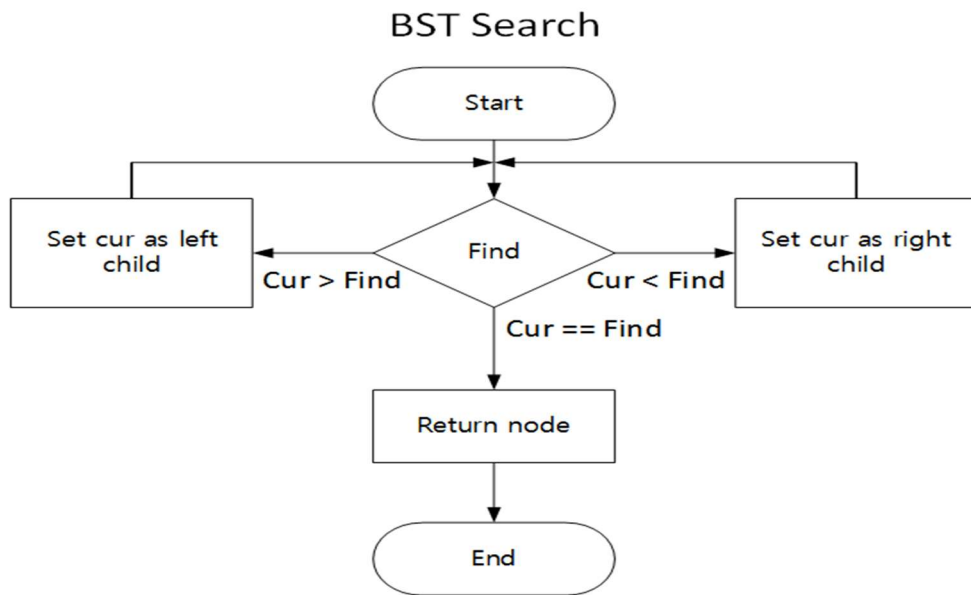
i. Insert



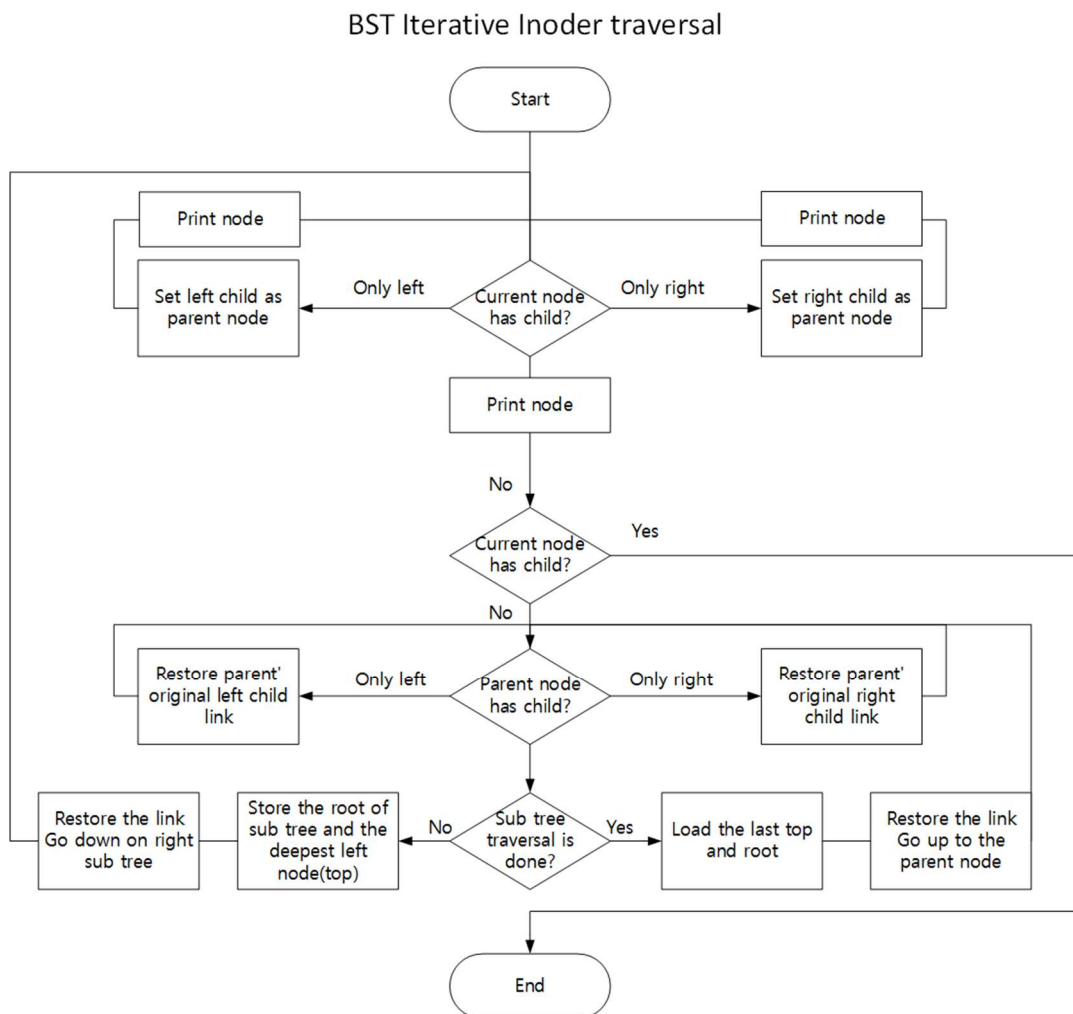
ii. Delete



iii. Search

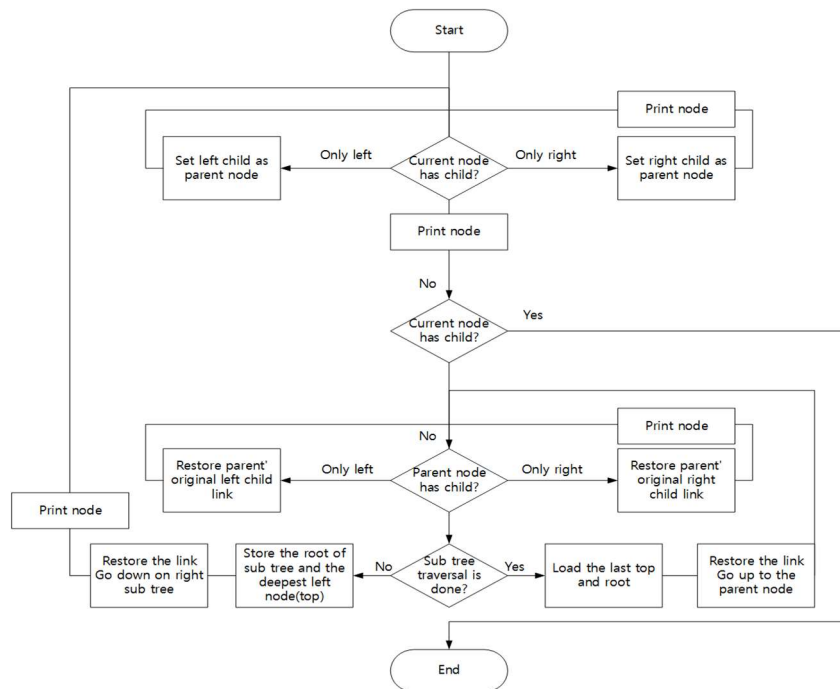


iv. Inorder



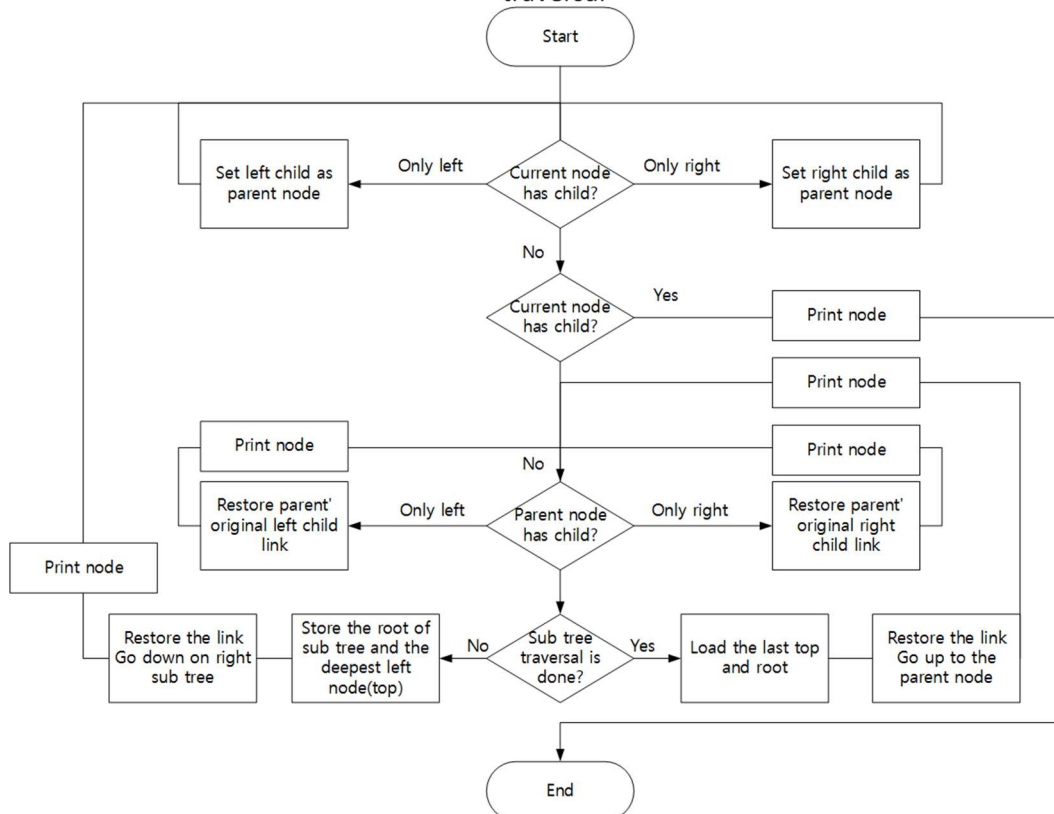
v. Preorder

BST Iterative Preoder traversal



vi. Postorder

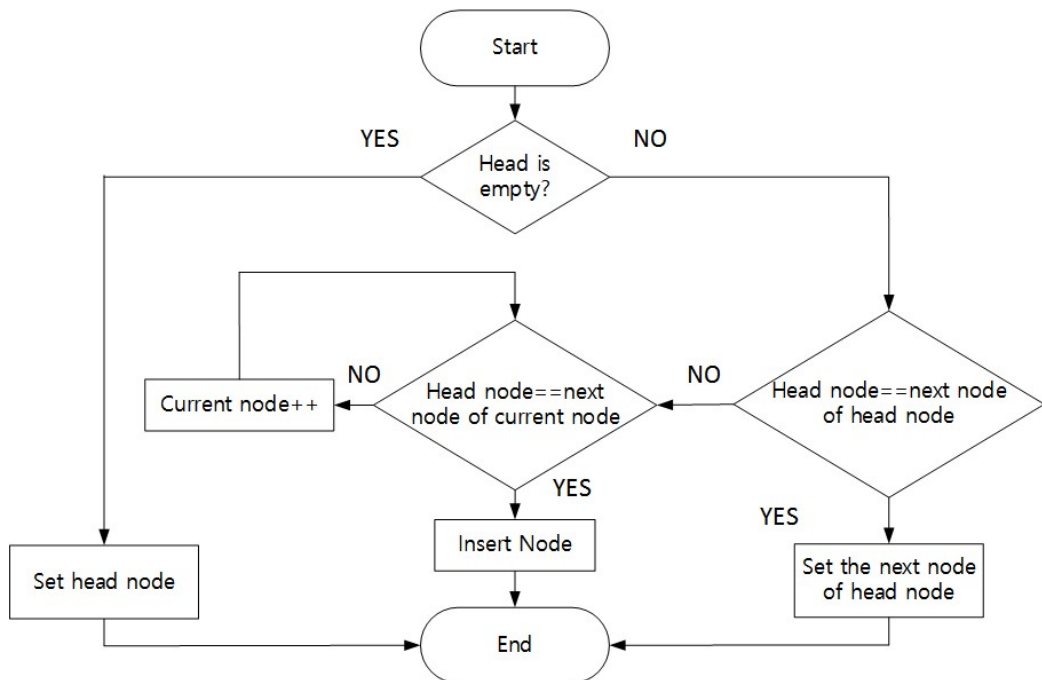
BST Iterative Postoder traversal



3) Circular Linked List

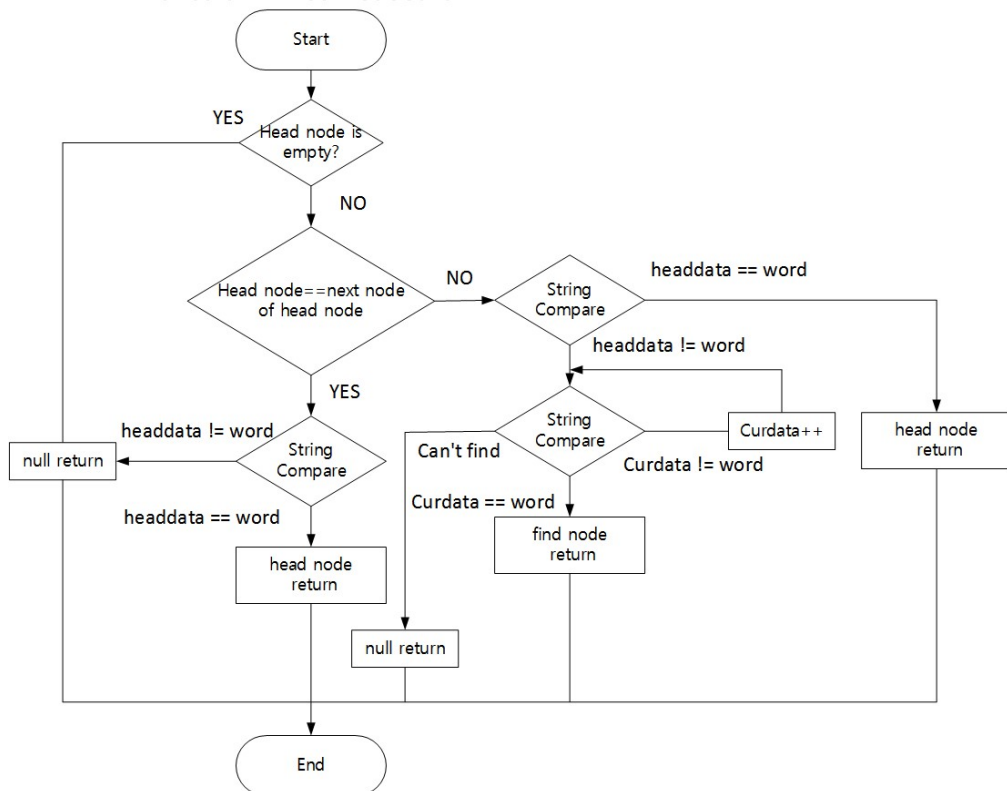
i) Insert

Circular Linked List Insert



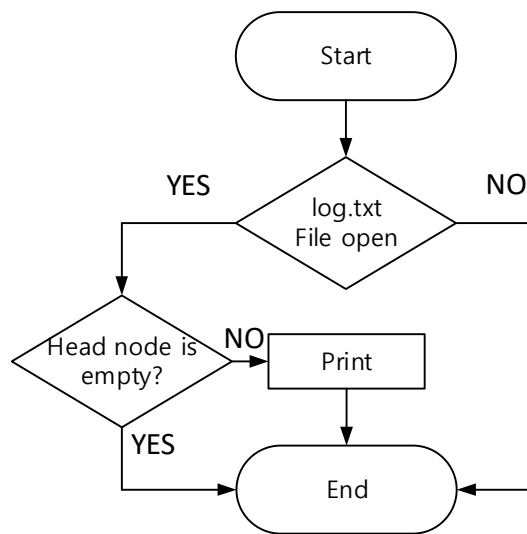
ii) Search

Circular Linked List Search



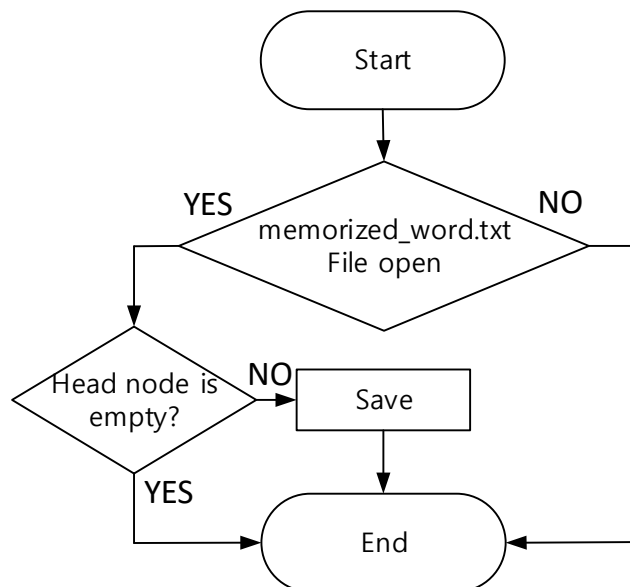
iii) Print

Circular Linked List Print



iv) Save

Circular Linked List Save



4. 결과화면

1) LOAD

- LOAD command

```
ubuntu@ubuntu-  
LOAD word.txt  
LOAD  
PRINT TO_MEMORIZE  
PRINT MEMORIZING I_IN  
PRINT MEMORIZED  
EXIT
```

- LOAD command 결과화면

```
ubuntu@ubuntu-Virtu  
===== ERROR =====  
100  
=====
```

```
===== LOAD =====  
Success  
=====
```

LOAD 뒤에 word.txt때문에 ERROR가 발생했고 ERROR CODE는 100이다.

두번째 LOAD의 실행은 성공했다.

- to_memorize_word.txt의 단어

```
ubuntu@ubuntu-Virtu  
famous 유명한  
special 특별한  
just 단지  
nature 자연  
restaurant 식당  
group 집단  
habit 습관  
culture 문화  
information 정보  
advertisement 광고
```

```
pole 막대기  
rubber 고무  
root 뿌리  
structure 구조  
vote 투표  
patient 환자  
quick 빠른  
captain 선장  
bucket 양동이  
cage 새장  
kite 연  
miracle 기적
```

```
1,1
```

```
248,1
```

총 248개의 단어가 있다.

- PRINT TO_MEMORIZE

```

===== PRINT =====
famous 유명한
special 특별한
just 단지
nature 자연
restaurant 식당
quick 빠른
captain 선장
bucket 양동이
cage 새장
kite 연
miracle 기적
=====

```

10,1 257,1

257-10+1=248개의 단어가 그대로 들어왔음을 볼 수 있다.

- memorizing_word.txt의 단어와 PRINT MEMORIZING

```

ubuntu@ubuntu-V
problem 문제
place 장소
part 부분
paper 종이
plan 계획
plant 식물
hand 손
different 다른
bird 새
autumn 가을
clothes 옷
clean 깨끗한
course 강좌
fire 불
face 얼굴
fun 재미
earth 지구
each 각각
enjoy 즐기다
life 인생
letter 편지
learn 배우다
lot 많이
listen 듣다
light 빛
job 직업
name 이름
newspaper 신문
make 만들다
movie 영화
mind 마음
trip 여행
story 이야기
spring 봄
space 공간
start 시작
summer 여름
street 거리
vacation 휴가
volunteer 자원봉사자
visit 방문
work 일
winter 겨울
watch 시계
win 이기다
~

```

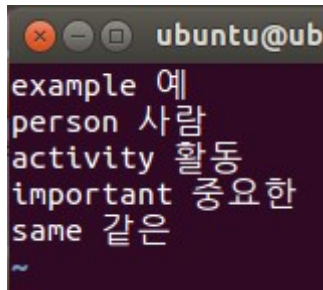
```

ubuntu@ubuntu-Virt
===== PRINT =====
autumn 가을
bird 새
clean 깨끗한
clothes 옷
course 강좌
different 다른
each 각각
earth 지구
enjoy 즐기다
face 얼굴
fire 불
fun 재미
hand 손
job 직업
learn 배우다
letter 편지
life 인생
light 빛
listen 듣다
lot 많이
make 만들다
mind 마음
movie 영화
name 이름
newspaper 신문
paper 종이
part 부분
place 장소
plan 계획
plant 식물
problem 문제
space 공간
spring 봄
start 시작
story 이야기
street 거리
summer 여름
trip 여행
vacation 휴가
visit 방문
volunteer 자원봉사자
watch 시계
win 이기다
winter 겨울
work 일
=====

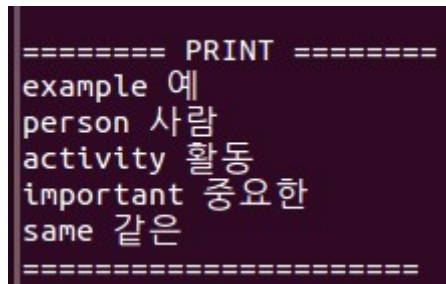
```

두 개 모두 45개의 단어를 가지고 있고 PRINT는 정렬된 형태도 나오는 것을 확인 할 수 있습니다.

- memorized_word.txt의 단어와 PRINT MEMORIZED



```
ubuntu@ubuntu-V:~$ cat memorized_word.txt
example 예
person 사람
activity 활동
important 중요한
same 같은
~
```

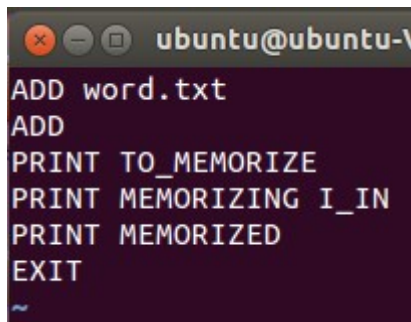


```
===== PRINT =====
example 예
person 사람
activity 활동
important 중요한
same 같은
=====
~
```

두 개 모두 5개의 단어를 가지고 있음을 확인할 수 있습니다.

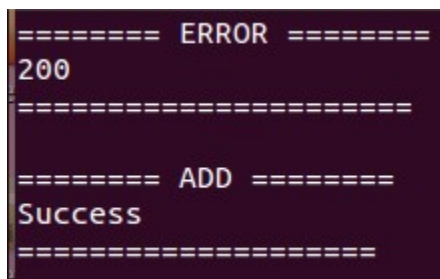
2) ADD

- ADD command



```
ubuntu@ubuntu-V:~$ ADD word.txt
ADD
PRINT TO_MEMORIZE
PRINT MEMORIZING I_IN
PRINT MEMORIZED
EXIT
~
```

- ADD command 결과화면



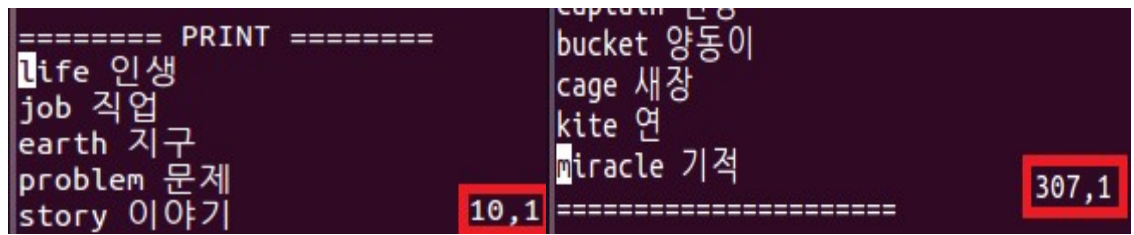
```
===== ERROR =====
200
=====

===== ADD =====
Success
=====
~
```

ADD 뒤에 word.txt가 있어 ERROR가 났으며 ERROR CODE는 200이다.

두 번째 ADD는 실행이 성공했다.

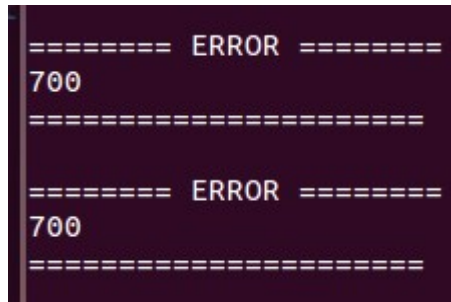
- PRINT TO_MEMORIZE



```
===== PRINT =====
life 인생
job 직업
earth 지구
problem 문제
story 이야기
bucket 양동이
cage 새장
kite 연
miracle 기적
=====
```

총 298개의 단어가 잘 넘어갔음을 볼 수 있다.

- PRINT MEMORIZING ,PRINT MEMORIZED



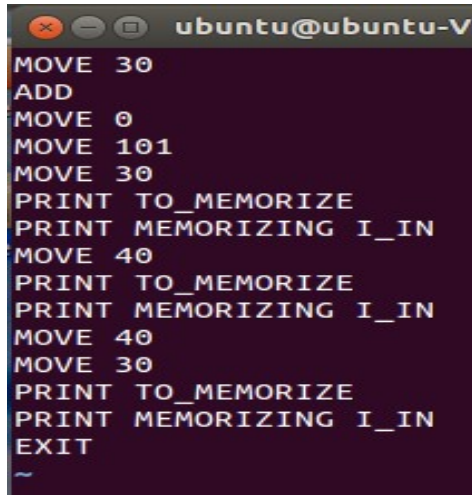
```
===== ERROR =====
700
=====

===== ERROR =====
700
=====
```

MEMORIZING , MEMORIZED 모두 데이터가 없으므로 ERROR CODE를 출력한다. PRINT의 ERROR CODE는 700이다.

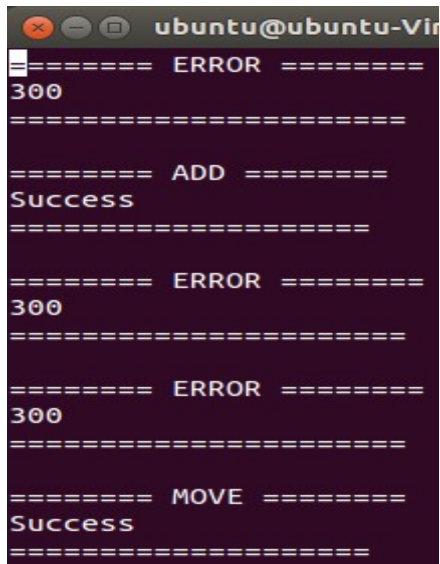
3) MOVE

- MOVE command



```
ubuntu@ubuntu-V
MOVE 30
ADD
MOVE 0
MOVE 101
MOVE 30
PRINT TO_MEMORIZE
PRINT MEMORIZING I_IN
MOVE 40
PRINT TO_MEMORIZE
PRINT MEMORIZING I_IN
MOVE 40
MOVE 30
PRINT TO_MEMORIZE
PRINT MEMORIZING I_IN
EXIT
```

- MOVE command 결과화면



```
ubuntu@ubuntu-V
===== ERROR =====
300
=====

===== ADD =====
Success
=====

===== ERROR =====
300
=====

===== ERROR =====
300
=====

===== MOVE =====
Success
=====
```

처음에 TO_MEMORIZE에 데이터가 하나도 없기 때문에 MOVE 30은 ERROR가 출력된다. ADD로 TO_MEMORIZE의 데이터가 들어왔지만 ADD 0 이나 100이 넘는 수는 ERROR CODE가 출력된다. 그래서 MOVE 30만 제대로 성공한다. MOVE의 ERROR CODE는 300이다.

- MOVE 30

■ TO_MEMORIZE

```
===== PRINT =====
vacation 휴가
summer 여름
course 강좌
spring 봄
autumn 가을
quick 빠른
captain 선장
bucket 양동이
cage 새장
kite 연
miracle 기적
=====
```

22,1 289,1

■ MEMORIZING

```
===== PRINT =====
activity 활동
bird 새
clothes 옷
different 다른
each 각각
earth 지구
example 예
fire 불
fun 재미
hand 손
important 중요한
job 직업
learn 배우다
letter 편지
life 인생
listen 듣다
lot 많이
make 만들다
movie 영화
name 이름
part 부분
person 사람
place 장소
plan 계획
plant 식물
problem 문제
same 같은
story 이야기
trip 여행
work 일
=====
===== MOVE =====
Success
=====
```

30개의 데이터가 TO_MEMORIZE에서 MEMORIZING으로 옮겨졌다.

- MOVE 40

■ TO_MEMORIZE

```
===== PRINT =====
always 언제나
history 역사
state 상태
soldier 군인
village 마을
office 사무실
quick 빠른
captain 선장
bucket 양동이
cage 새장
kite 연
miracle 기적
=====
```

330,1 557,1

■ MEMORIZING

<pre> ===== PRINT ===== activity 활동 advertisement 광고 autumn 가을 back 뒤 bird 새 </pre>	<pre> volunteer 자원봉사자 war 전쟁 watch 시계 win 이기다 winter 겨울 Work 일 ===== ===== ERROR ===== 300 ===== ===== MOVE ===== Success ===== </pre>
---	--

40개의 단어가 더 넘어왔음을 확인 할 수 있다.

- MOVE 40 and MOVE 30

■ TO_MEMORIZE

<pre> ===== PRINT ===== fact 사실 rule 규칙 die 죽다 busy 바쁜 sick 아픈 health 건강 holiday 공휴일 goal 목표 order 순서 </pre>	<pre> captain 선장 bucket 양동이 cage 새장 kite 연 miracle 기적 ===== </pre>
--	--

■ MEMORIZING

<pre> ===== PRINT ===== activity 활동 advertisement 광고 always 언제나 area 지역 art 미술 </pre>	<pre> war 전쟁 watch 시계 wear 입다 win 이기다 winter 겨울 Work 일 ===== </pre>
---	---

70개의 데이터가 더 넘어왔음을 확인할 수 있다.

4) TEST

- TEST command and 결과화면

```
ubuntu@ubuntu-V
TEST example 예
TEST Example 예
ADD
MOVE 50
TEST
TEST example 예
TEST Person 사람
TEST SAME 같은
PRINT TO_MEMORIZE
PRINT MEMORIZING I_IN
PRINT MEMORIZED
EXIT
```

```
ubuntu@ubuntu-V
===== ERROR =====
500
===== ERROR =====
500
===== ADD =====
Success
===== MOVE =====
Success
===== ERROR =====
500
===== TEST =====
Pass
===== TEST =====
Pass
===== TEST =====
Pass
```

아무 데이터가 없어서 처음에 ERROR가 발생한다. ADD MOVE이후에는 TEST 뒤에 단어와 뜻이 없으므로 ERROR가 발생하고 다른 경우에는 성공하여 pass가 출력되는 것을 볼 수 있다. TEST의 ERROR CODE는 500이다.

- PRINT MEMORIZED

```
===== PRINT =====
example 예
person 사람
same 같은
=====
```

TEST에 의해서 3개의 단어가 MEMORIZED로 옮겨진 것을 볼 수 있다.

5) SAVE

- SAVE command와 결과화면

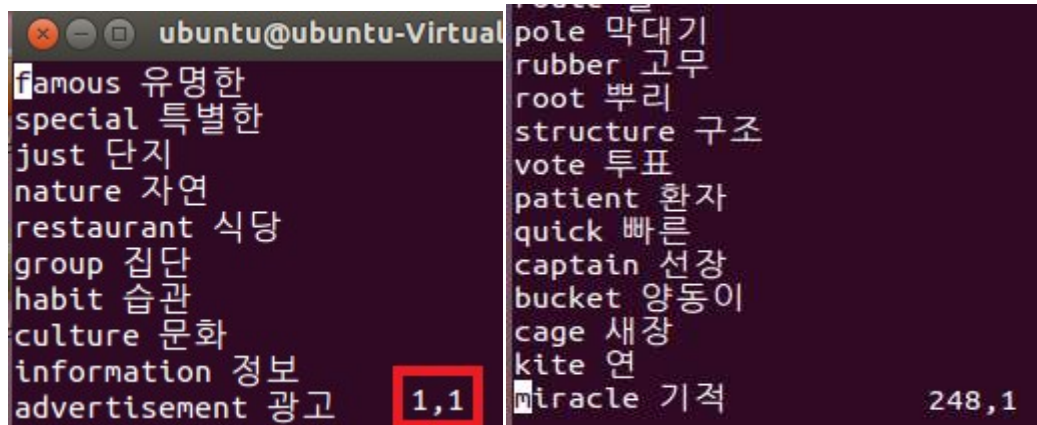
```
ubuntu@ubuntu-V
SAVE
ADD
MOVE 50
TEST example 예
TEST Person 사람
TEST activity 활동
TEST important 중요한
TEST SAME 같은
SAVE
EXIT
```

```
ubuntu@ubuntu-V
===== ERROR =====
400
===== ERROR =====
400
===== ADD =====
Success
===== MOVE =====
Success
===== TEST =====
Pass
===== TEST =====
Pass
===== TEST =====
Pass
===== TEST =====
Pass
===== TEST =====
Pass
===== SAVE =====
Success
```

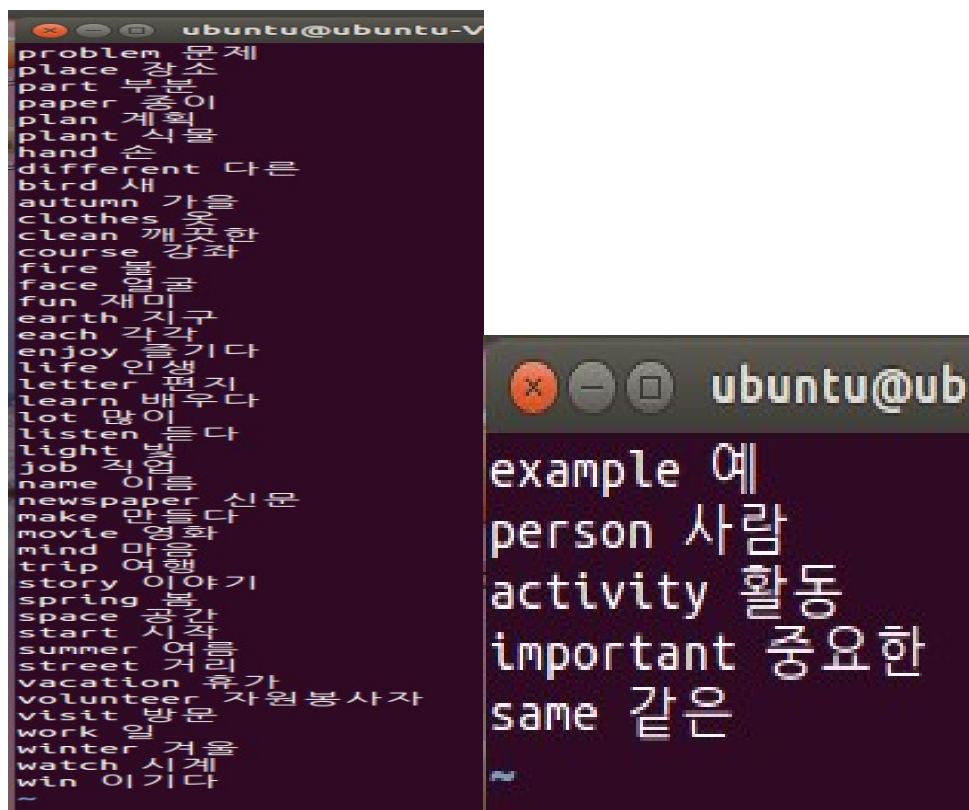
처음에 SAVE할 데이터가 없기 때문에 ERROR가 발생함을 볼 수 있다. SAVE의 ERROR CODE는 400이다.

- SAVE txt

■ to_memorize_word.txt



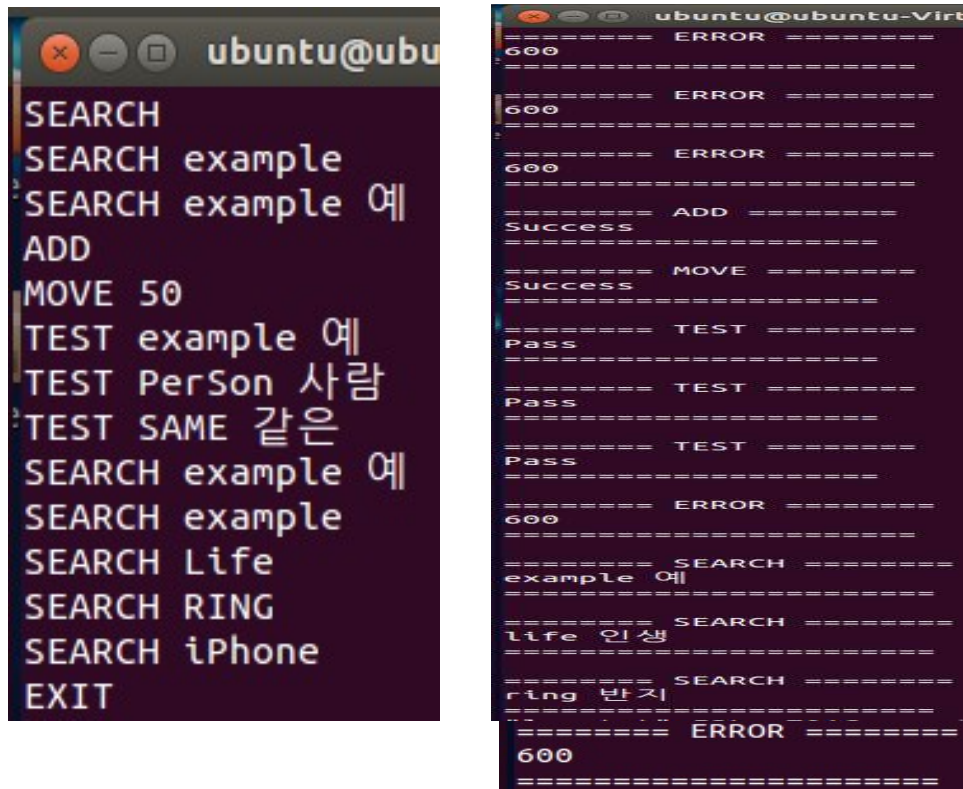
■ memorizing_word.txt and memorized_word.txt



MOVE 50에서 TEST에 의해서 5개의 단어가 정확히 이동한 것을 확인 할 수 있다.

6) SEARCH

- SEARCH command and 결과화면

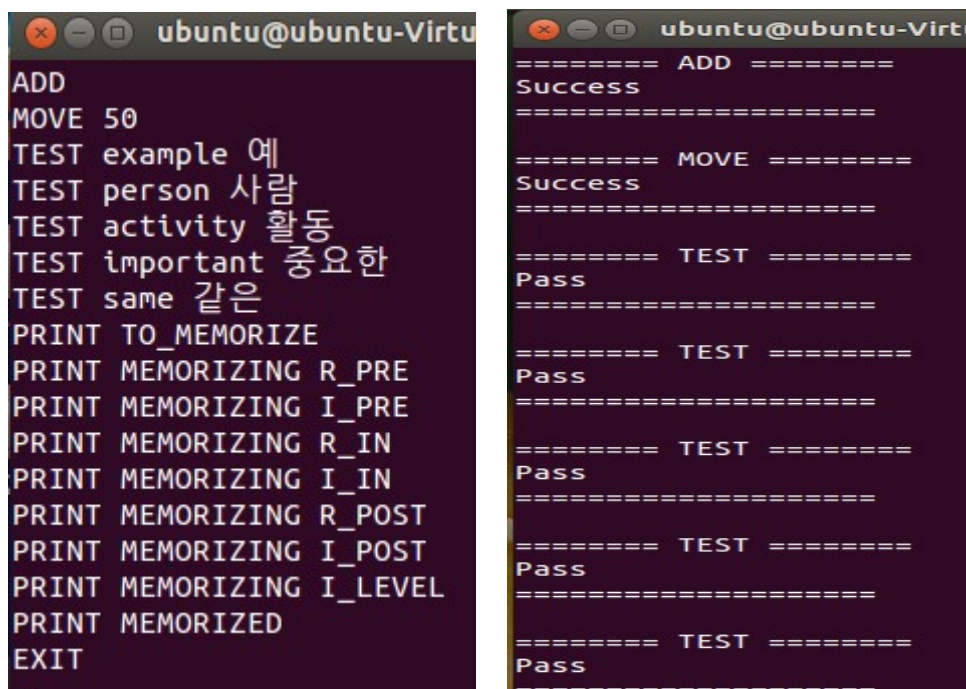


The left screenshot shows a terminal window with the following commands: SEARCH, SEARCH example, SEARCH example 예, ADD, MOVE 50, TEST example 예, TEST Person 사람, TEST SAME 같은, SEARCH example 예, SEARCH example, SEARCH Life, SEARCH RING, SEARCH iPhone, and EXIT. The right screenshot shows the output of these commands, including several ERROR messages with code 600 and SUCCESS/Pass messages for the ADD, MOVE, and TEST commands.

처음 데이터가 없기에 ERROR가 출력 됨을 볼 수 있으며 ADD, MOVE, TEST 이후에는 SEARCH뒤에 단어만 와야 한다. 그렇지 않은 경우에는 ERROR가 출력 되는 것을 볼 수 있다. 마지막 iphone은 없는 단어 이므로 ERROR CODE가 출력된다. SEARCH의 ERROR CODE는 600이다.

7) PRINT

- PRINT command and 결과화면



The left screenshot shows a terminal window with the following commands: ADD, MOVE 50, TEST example 예, TEST person 사람, TEST activity 활동, TEST important 중요한, TEST same 같은, PRINT TO_MEMORIZE, PRINT MEMORIZING R_PRE, PRINT MEMORIZING I_PRE, PRINT MEMORIZING R_IN, PRINT MEMORIZING I_IN, PRINT MEMORIZING R_POST, PRINT MEMORIZING I_POST, PRINT MEMORIZING I_LEVEL, PRINT MEMORIZED, and EXIT. The right screenshot shows the output of these commands, including several SUCCESS and Pass messages for the ADD, MOVE, and TEST commands, and several PRINT MEMORIZING messages for the PRINT commands.

- PRINT R_PRE and PRINT I_PRE

```
ubuntu@ubuntu-Virt
===== PRINT =====
problem 문제
place 장소
part 부분
paper 종이
plan 계획
plant 식물
hand 손
different 다른
bird 새
autumn 가을
clothes 옷
clean 깨끗한
course 강좌
fire 불
face 얼굴
fun 재미
earth 지구
each 각각
enjoy 즐기다
life 인생
letter 편지
learn 배우다
lot 많이
listen 듣다
light 빛
job 직업
name 이름
newspaper 신문
make 만들다
movie 영화
mind 마음
trip 여행
story 이야기
spring 봄
space 공간
start 시작
summer 여름
street 거리
vacation 휴가
volunteer 자원봉사자
visit 방문
work 일
winter 겨울
watch 시계
win 이기다
=====
```

```
ubuntu@ubuntu-Virt
===== PRINT =====
problem 문제
place 장소
part 부분
paper 종이
plan 계획
plant 식물
hand 손
different 다른
bird 새
autumn 가을
clothes 옷
clean 깨끗한
course 강좌
fire 불
face 얼굴
fun 재미
earth 지구
each 각각
enjoy 즐기다
life 인생
letter 편지
learn 배우다
lot 많이
listen 듣다
light 빛
job 직업
name 이름
newspaper 신문
make 만들다
movie 영화
mind 마음
trip 여행
story 이야기
spring 봄
space 공간
start 시작
summer 여름
street 거리
vacation 휴가
volunteer 자원봉사자
visit 방문
work 일
winter 겨울
watch 시계
win 이기다
=====
```

- PRINT R_IN and PRINT I_IN


```

ubuntu@ubuntu-Virt
===== PRINT =====
autumn 가을
bird 새
clean 깨끗한
clothes 옷
course 강좌
different 다른
each 각각
earth 지구
enjoy 즐기다
face 얼굴
fire 불
fun 재미
hand 손
job 직업
learn 배우다
letter 편지
life 인생
light 빛
listen 듣다
lot 많이
make 만들다
mind 마음
movie 영화
name 이름
newspaper 신문
paper 종이
part 부분
place 장소
plan 계획
plant 식물
problem 문제
space 공간
spring 봄
start 시작
story 이야기
street 거리
summer 여름
trip 여행
vacation 휴가
visit 방문
volunteer 자원봉사자
watch 시계
win 이기다
winter 겨울
work 일
=====

```

```

ubuntu@ubuntu-Virt
===== PRINT =====
autumn 가을
bird 새
clean 깨끗한
clothes 옷
course 강좌
different 다른
each 각각
earth 지구
enjoy 즐기다
face 얼굴
fire 불
fun 재미
hand 손
job 직업
learn 배우다
letter 편지
life 인생
light 빛
listen 듣다
lot 많이
make 만들다
mind 마음
movie 영화
name 이름
newspaper 신문
paper 종이
part 부분
place 장소
plan 계획
plant 식물
problem 문제
space 공간
spring 봄
start 시작
story 이야기
street 거리
summer 여름
trip 여행
vacation 휴가
visit 방문
volunteer 자원봉사자
watch 시계
win 이기다
winter 겨울
work 일
=====

```

- PRINT R_POST and PRINT I_POST

```

ubuntu@ubuntu-Virt
===== PRINT =====
autumn 가을
clean 깨끗한
course 강좌
clothes 옷
bird 새
each 각각
enjoy 즐기다
earth 지구
face 얼굴
fun 재미
fire 불
different 다른
job 직업
mind 마음
movie 영화
make 만들다
newspaper 신문
name 이름
learn 배우다
letter 편지
light 빛
listen 듣다
lot 많이
life 인생
hand 손
space 공간
start 시작
spring 봄
street 거리
summer 여름
story 이야기
win 이기다
watch 시계
winter 겨울
work 일
visit 방문
volunteer 자원봉사자
vacation 휴가
trip 여행
paper 종이
part 부분
plant 식물
plan 계획
place 장소
problem 문제
=====

```

```

ubuntu@ubuntu-Virt
===== PRINT =====
autumn 가을
clean 깨끗한
course 강좌
clothes 옷
bird 새
each 각각
enjoy 즐기다
earth 지구
face 얼굴
fun 재미
fire 불
different 다른
job 직업
mind 마음
movie 영화
make 만들다
newspaper 신문
name 이름
learn 배우다
letter 편지
light 빛
listen 듣다
lot 많이
life 인생
hand 손
space 공간
start 시작
spring 봄
street 거리
summer 여름
story 이야기
win 이기다
watch 시계
winter 겨울
work 일
visit 방문
volunteer 자원봉사자
vacation 휴가
trip 여행
paper 종이
part 부분
plant 식물
plan 계획
place 장소
problem 문제
=====

```


- PRINT I_LEVEL

```

===== PRINT =====
problem 문제
place 장소
part 부분
plan 계획
paper 종이
plant 식물
hand 손
different 다른
life 인생
letter 편지
lot 많이
learn 배우다
listen 듣다
light 빛
trip 여행
bird 새
fire 불
face 얼굴
fun 재미
job 직업
name 이름
newspaper 신문
vacation 휴가
volunteer 자원봉사자
visit 방문
autumn 가을
clothes 옷
clean 깨끗한
course 강좌
earth 지구
each 각각
enjoy 즐기다
make 만들다
movie 영화
mind 마음
story 이야기
spring 봄
summer 여름
space 공간
start 시작
street 거리
work 일
winter 겨울
watch 시계
win 이기다
=====

```

8) UPDATE

- UPDATE command and 결과화면

```

ADD
MOVE 50
TEST example 예
TEST person 사람
TEST activity 활동
TEST important 중요한
TEST same 같은
UPDATE
UPDATE example
UPDATE example 예시
UPDATE Life 생애
UPDATE RING 링
PRINT TO_MEMORIZE
PRINT MEMORIZING I_IN
PRINT MEMORIZED
EXIT

```

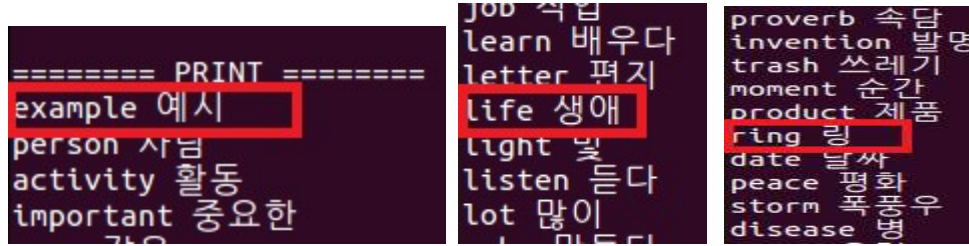
```

===== ADD =====
Success
=====
===== MOVE =====
Success
=====
===== TEST =====
Pass
=====
===== TEST =====
Pass
=====
===== TEST =====
Pass
=====
===== TEST =====
Pass
=====
===== TEST =====
Pass
=====
===== ERROR =====
800
=====
===== ERROR =====
800
=====
===== UPDATE =====
example 예 -> 예시
=====
===== UPDATE =====
life 인생 -> 생애
=====
===== UPDATE =====
ring 반지 -> 링
=====

```

UPDATE는 뒤에 단어와 뜻을 모두 써야 합니다. 그리하여 처음 두 경우에는 ERROR가 발생하는 것을 볼 수 있습니다. UPDATE의 ERROR CODE는 800입니다.

- UPDATE 결과화면



9) EXIT

- 위의 결과들에서 잘 됨을 확인 할 수 있다.

5. 고찰

김병휘: Manager.cc, MEMORIZING I_PRE, I_IN, I_POST (점수 :9.5)

팀 프로젝트를 진행하면서 혼자 하는 것이 아닌 서로 소통하는 방법을 배웠고 이를 통해 다른 학생들의 아이디어나 노하우를 얻을 수 있는 시간이 되었습니다.

해당 프로젝트에서는 Manager 클래스의 함수와 Traversal 함수를 담당하였습니다. 특히 Traversal 함수를 구현하는 데 있어서 Recursive한 순회는 비교적 간단하나 iterative한 방법을 구현하는 데 조금 어려웠습니다. 그 가운데서도 스택을 사용하는 방법과 사용하지 않는 방법이 있는데 해당 프로젝트에서는 사용하지 않는 방법을 사용하였습니다. 핵심적인 방법은 현재 노드를 방문하며 현재 노드는 부모 노드를 가리키도록 하고 본래의 자식 노드로 내려가며 순회를 하는 방법이었습니다. 이를 통하여 자료 구조에 대한 조금 더 폭 넓은 시야를 갖게 되었고 나중에 코딩하는 데에 있어서 많은 도움이 될 것이라 생각하게 되었습니다.

조원희: TO_MEMORIZE Queue, MEMORIZED Circular Linked List(점수 :8)

팀 프로젝트를 수행하면서 혼자서 했다면 오래 고민하고 막혔을 부분에서 질문과 토의를 통해 문제점을 해결해 나갈 수 있어서 좋았습니다.

해당 프로젝트에서는 Queue 구조체와 Circular Linked List를 구현하는 부분을 담당하였습니다. Queue를 만드는데 있어서 문제는 없었지만 다음 자료구조로 넘어간다는 생각을 하지 못하여 Pop함수에서 return이 아닌 delete를 하는 실수를 가졌습니다. 또한 Circular Linked List를 구현함에 있어서는 처음부터 끝 노드의 next를 head로 연결하도록 만들었는데 이 부분에서 노드가 하나만 들어있는 경우에 대해서도 예외처리를 해주어야 했는데 해주지 않아서 오류가 생겼었지만 디버깅모드를 이용하여 문제를 해결하였습니다.

김정환: MEMORIZING BST(Alphabet BST, Word BST) (점수 :7)

편입 후 팀 프로젝트가 처음이었습니다. 조원들이 막히는 부분이 있으면 도와주고 제가 이해가 되지 않거나 잘못 알고있는 부분에 대해서 설명을 자세하게 해주어 프로젝트에 대한 이해에 많은 도움이 되었습니다.

해당 프로젝트에서는 Alphabet BST와 Word BST를 구현하는 부분을 담당하였습니다. BST같은 경우는 강의자료와 교재를 참고하여 수월하게 구현하였습니다. Alphabet BST는 대문자로 이루어진 알파벳이 Level order의 트리 구조로 되어 있는데 Search함수에 대소문자 예외처리를 안해주어 오류가 생겼지만 초기에 발견하여 금방 해결하였습니다. Delete함수를 구현할 때 삭제 후 Circular Linked List에 넘겨주었지만 NULL값이 들어가 디버깅으로 확인해보니 노드의 데이터가 잘못 전달되는 것을 확인하고 오류를 수정하였습니다. 이번 프로젝트를 통하여 자료구조의 이해에 많은 도움이 되었습니다.