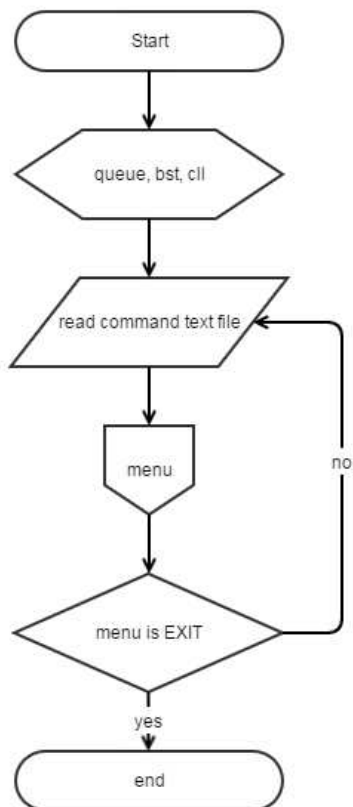


## 1. Introduction

이번 프로젝트는 영어 단어장을 큐, 이진트리, 환형 연결리스트를 이용하여 구현하는 것이다. 큐에서는 외워야할 단어들이 포함되어있고, 이진트리에는 외우고 있는 단어들(100개를 넘지 않는다), 환형 연결리스트에는 외운 단어들이 있다. 여기서 이진트리의 구조는 알파벳 노드로 구성되어있으며, 각 노드들은 그 알파벳으로 시작하는 단어들의 모음인 단어 이진트리를 갖게 된다.

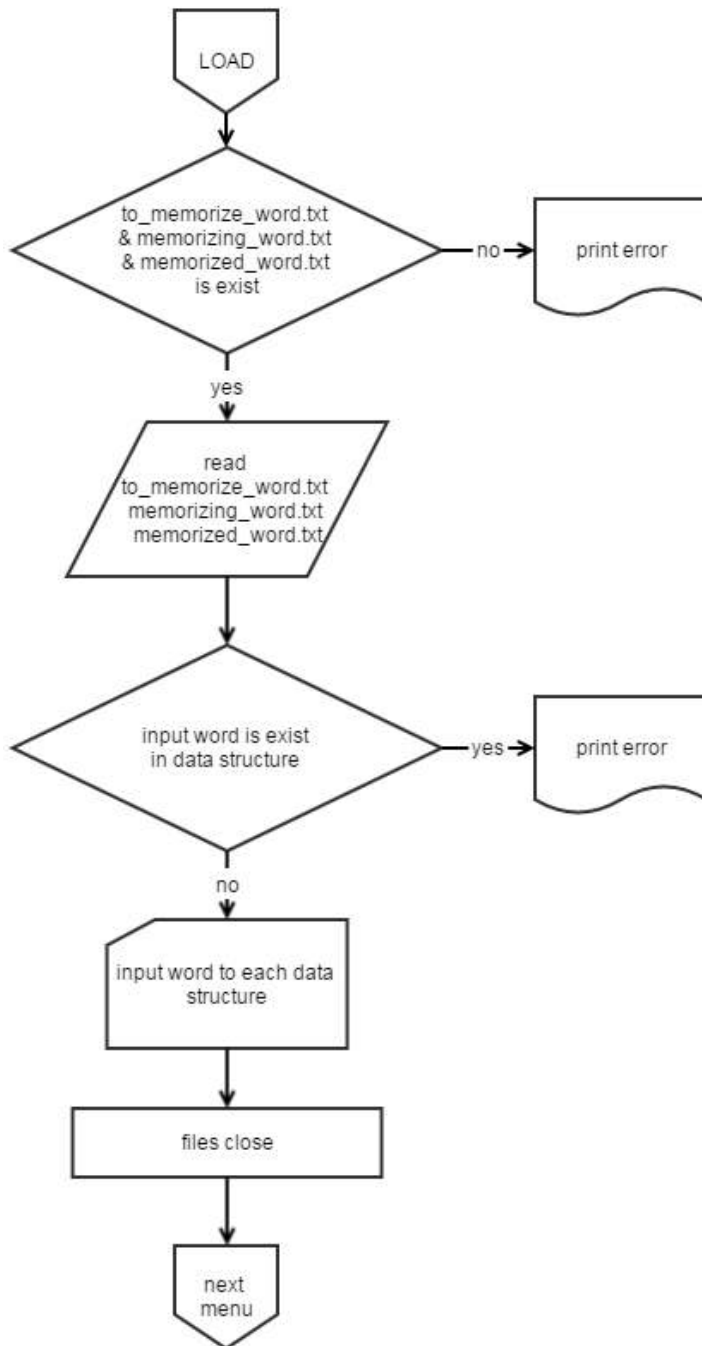
단어장 프로그램에는 LOAD, ADD, MOVE, SAVE, TEST, SEARCH, PRINT, UPDATE, EXIT 메뉴가 있다. LOAD는 기존의 단어장 파일을 불러오는 메뉴로, 세 파일이 모두 존재하지 않거나 이미 자료구조에 데이터가 있을 경우에 에러 코드를 출력한다. ADD는 단어 파일을 읽어 큐에 단어들을 추가하는 메뉴이다. 이때 해당 단어 파일이 존재 하지 않거나 단어가 없을 때 에러코드를 출력한다. MOVE는 큐의 단어들을 이진트리로 이동시키는 메뉴이다. 이때 옮길 단어의 수와 현재 이진트리에 존재하는 단어수의 합이 100을 넘을 수 없다. 큐에 단어가 존재 하지 않을 때에도 에러코드를 출력한다. SAVE는 현재 단어장의 정보를 저장하는 메뉴이다. TEST는 이진트리에서 환형 연결 리스트로 단어를 이동시킨다. 이진트리에 단어가 없거나 단어의 뜻이 틀리면 에러코드를 출력한다. SEARCH는 단어 검색하여 뜻을 찾아주는 명령어이다. 이때 검색범위는 큐, 이진트리, 환형 연결리스트 모두 포함된다. 찾는 단어가 없을 경우 에러코드를 출력한다. PRINT는 해당 자료구조의 단어를 출력하는 메뉴이다. UPDATE는 단어의 뜻을 바꿔주는 메뉴로, 이때도 찾는 단어가 없거나 자료구조에 단어가 없을 때 에러코드를 출력한다. EXIT은 프로그램을 종료시킨다.

## 2. Flowchart

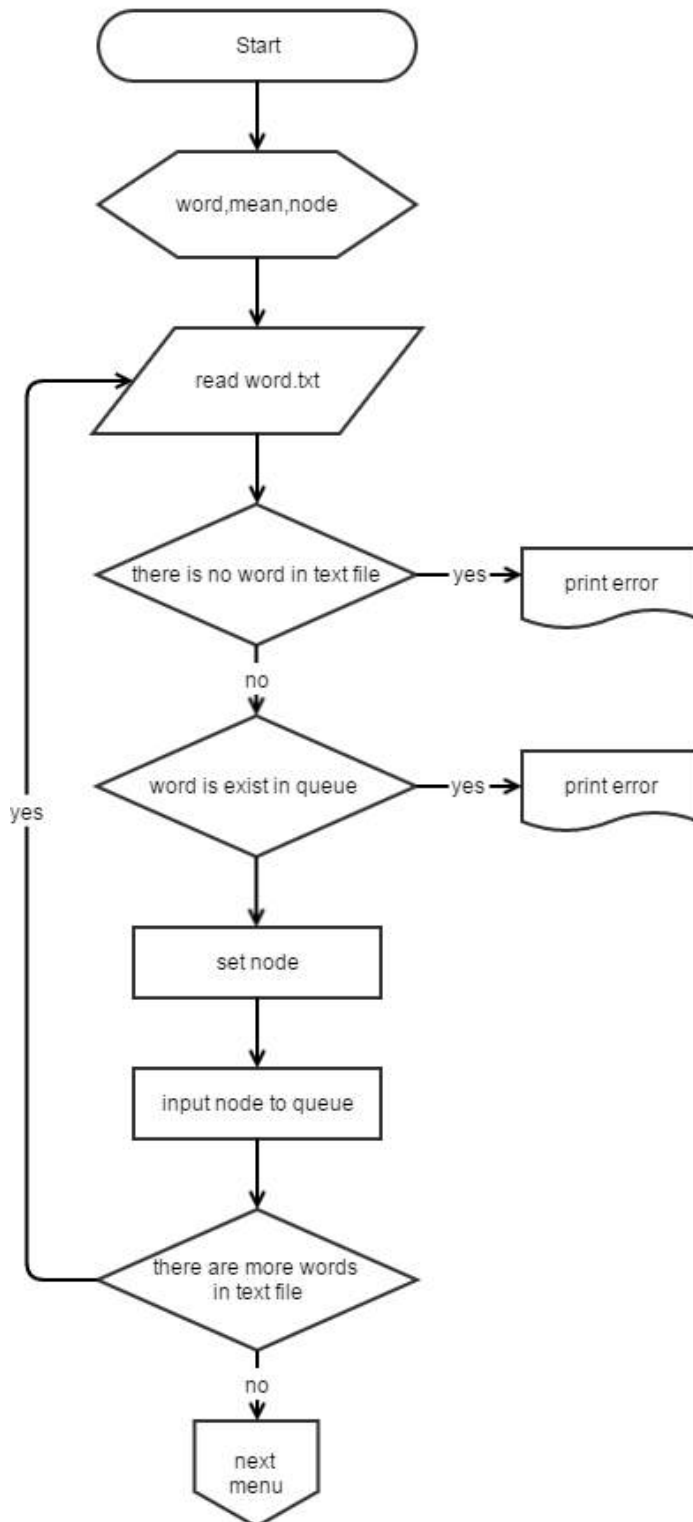


각 메뉴에 대한 플로우 차트는 다음과 같다.

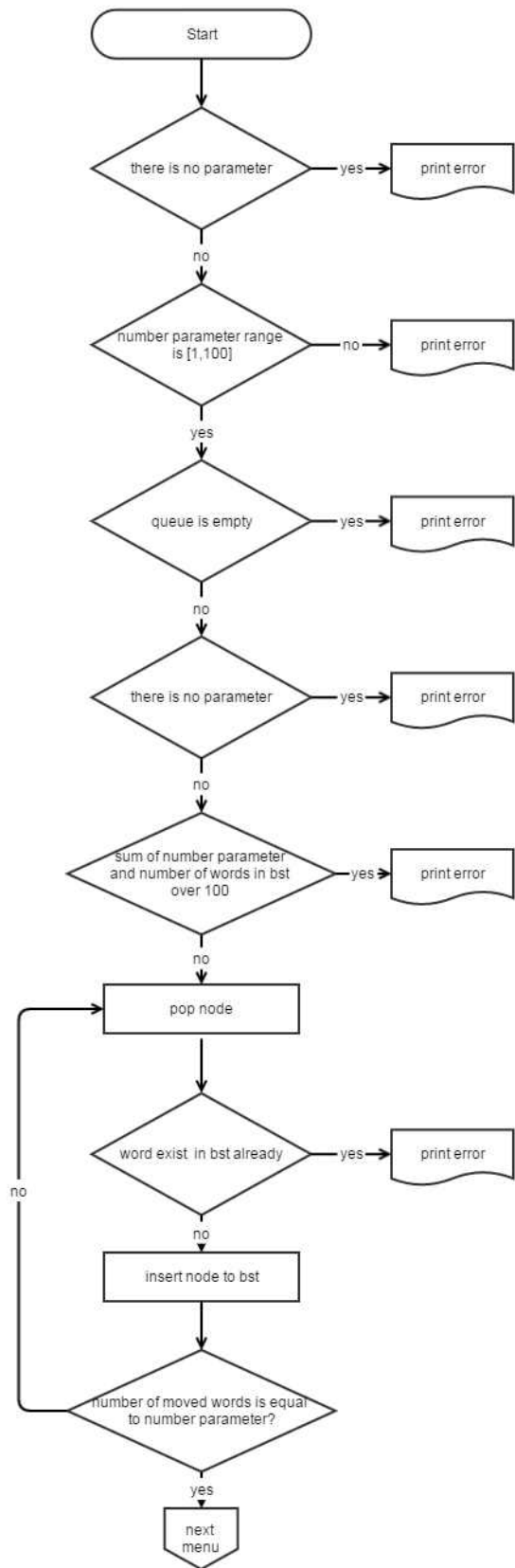
- LOAD



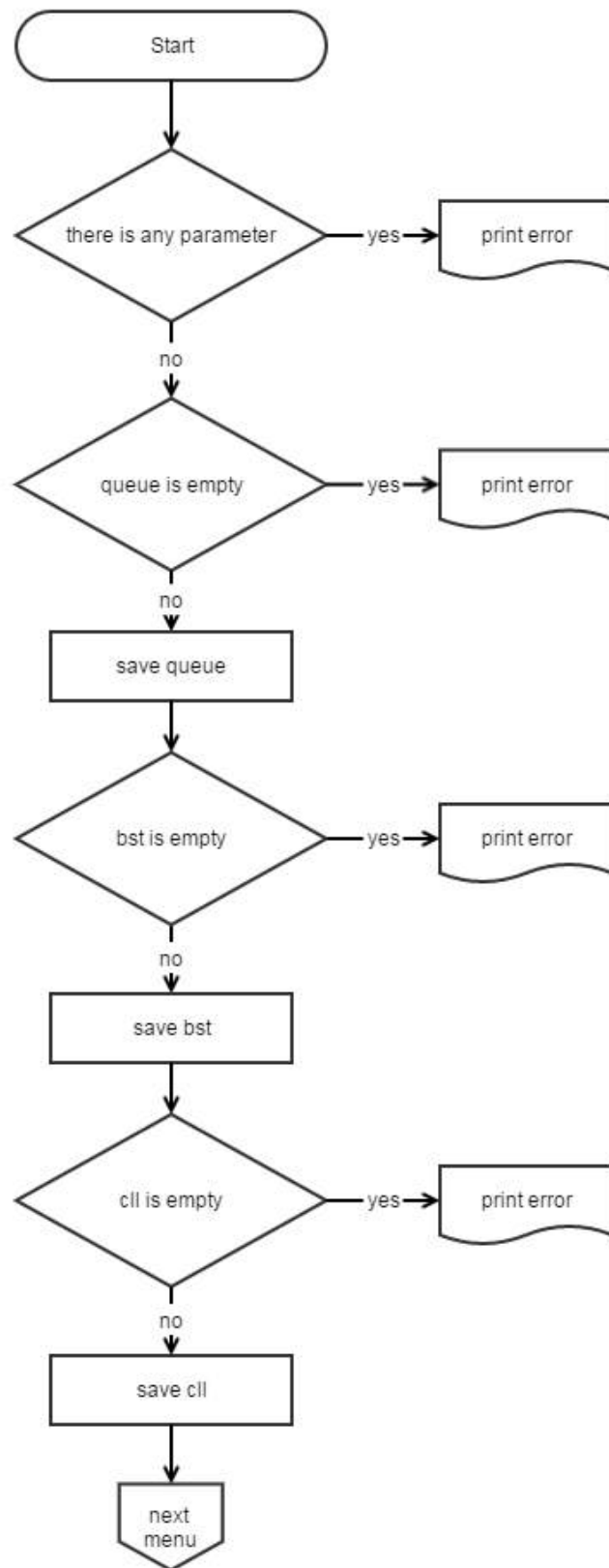
- ADD



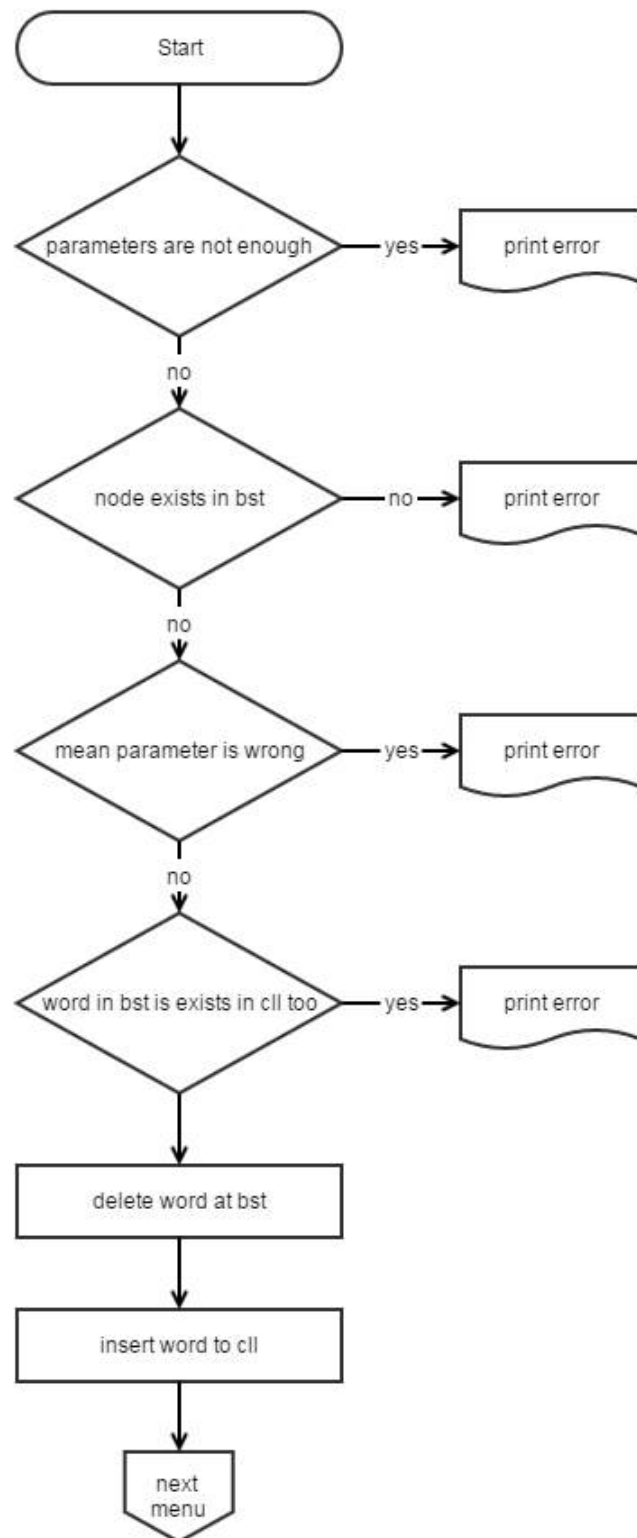
- MOVE



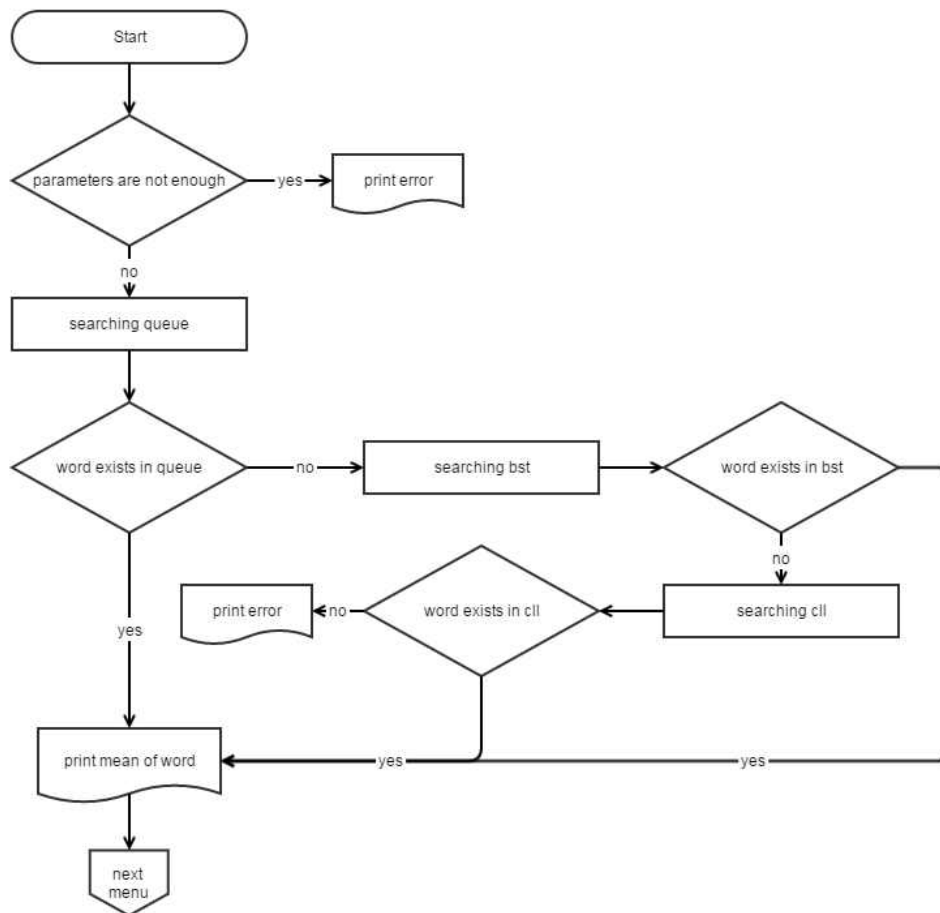
- SAVE



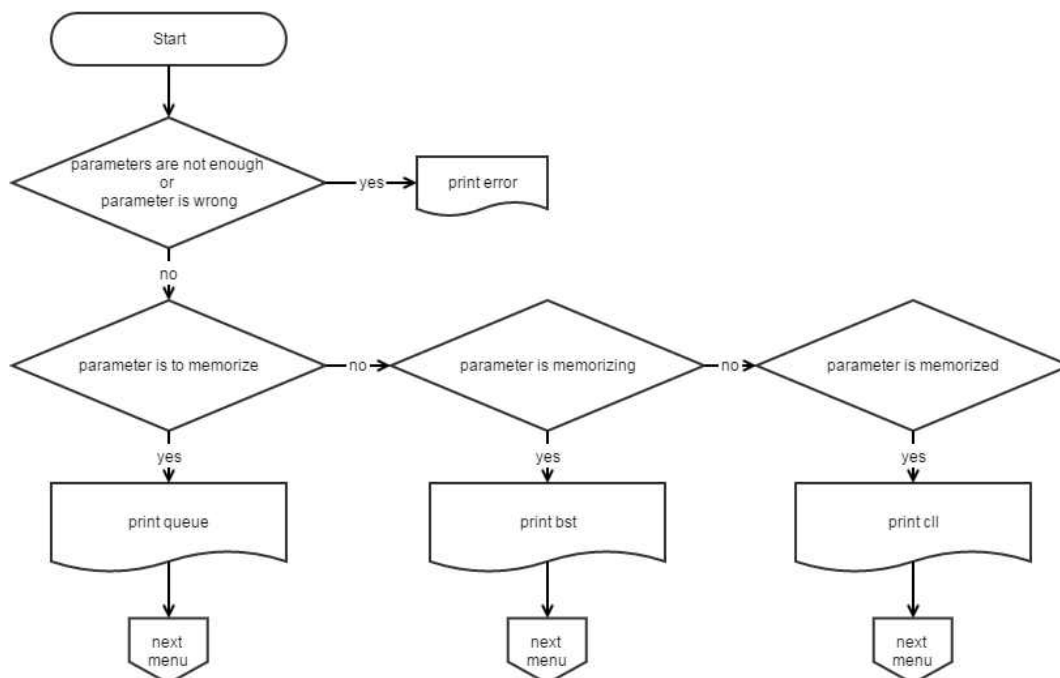
- TEST



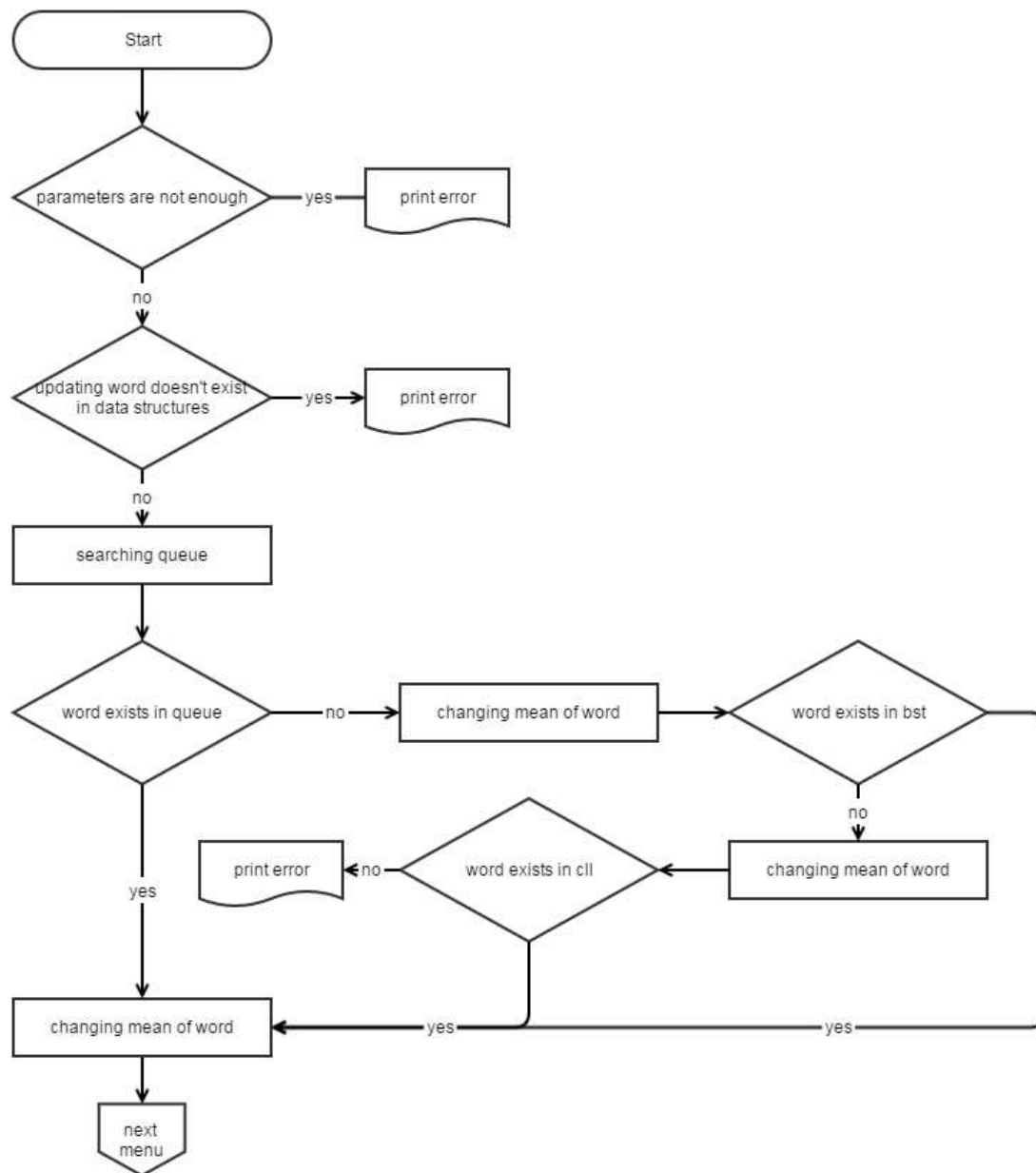
## - SEARCH



## - PRINT



- UPDATE





### 3. Algorithm

#### - Queue

: 먼저 들어온 노드가 먼저 나가는 FIFO형식의 자료구조이다. PUSH를 통해 큐에 노드가 들어오면 가장 마지막 노드의 다음에 추가된다. POP을 통해 노드가 나갈 때에는 head에 있는 노드를 큐에서 제거한다.

#### - BST

##### 1) Inset

: 이진트리에 root가 없는 경우엔 처음 들어오는 노드가 root가 된다. 그 이후엔 root를 기준으로 작은 노드는 왼쪽 자식으로, 큰 노드는 오른쪽 자식이 된다. 그 이후엔 root에서부터 추가되는 노드와 비교하여 작으면 왼쪽, 크면 오른쪽으로 이동하여 현재 이진트리의 leaf에 도달할 때까지 비교한다. 같은 노드를 발견하면, 추가하지 않고 때때 따라 에러코드를 출력한다.

##### 2) Print

###### (1) Pre-order

: 전위 순회는 parent->left child->right child 순으로 노드를 출력한다. 재귀함수와 반복문으로 구현가능하며 반복문 이용 시 스택을 이용한다. 스택은 FILO형식으로 먼저 들어온 노드가 나중에 출력되므로 parent를 먼저 출력하고 right child, left child를 스택에 추가하고 pop한 노드를 parent로 삼고 스택이 비어 있을 때까지 반복한다.

###### (2) In-order

: 중위 순회는 left child->parent->right child 순으로 노드를 출력한다. pre-order와 마찬가지로 재귀함수와 반복문으로 구현가능하며 반복문 이용 시 스택을 이용한다. 루트에서 왼쪽으로 이동하면서 스택을 노드에 추가하고 leaf에 도달했을 때 pop하고 pop된 노드를 출력하고 그 오른쪽 노드를 parent로 삼고 이를 반복한다.

###### (3) Post-order

: 후위 순회는 left child->right child->parent 순으로 노드를 출력한다. pre-order, in-order와 마찬가지로 재귀함수와 반복문으로 구현가능하며 반복문 이용 시 스택을 이용한다. 루트에서 가장 왼쪽으로 이동하여 스택에 노드를 추가하고 스택이 빌 때까지 스택에서 pop한 노드의 오른쪽 자식이 없으면 자신을 출력하고, 있으면 오른쪽 자식을 출력하는 방법을 반복한다.

###### (4) Level-order

: 낮은 레벨 순서대로 왼쪽에서 오른쪽으로 노드를 출력한다. 재귀함수와 반복문으로 구현가능하며 반복문 이용 시 큐를 이용한다. parent노드를 먼저 출력하고, left child를 push하고 right child를 push하고 pop한 노드를 다시 parent로 삼는 것을 반복한다.

##### 3) Search

: root에서부터 찾고자하는 노드를 비교하여 작으면 왼쪽 자식으로, 크면 오른쪽 자식으로 이동하면서 같은 노드를 발견할 때 까지 반복하는 방법을 이용한다.

##### 4) Delete

: 삭제하고자하는 노드를 search를 통해 찾고, 해당 노드가 left인 경우, left child만 있을 경우, right child만 있을 경우, 자식이 둘 다 있을 경우로 나누어 구현한다. 먼저 삭

제노드가 leaf인 경우, 부모가 없을 땐 root를 삭제한다. 부모의 왼쪽 자식이 삭제노드일 때 부모의 왼쪽 자식을 NULL로 설정한다. 오른쪽도 마찬가지이다. 삭제노드가 오른쪽 자식만 있을 경우, 부모가 없을 땐 삭제노드를 삭제노드가 root이므로 삭제노드의 오른쪽 자식을 root로 설정한다. 부모의 왼쪽자식이 삭제노드일 때에는 부모의 왼쪽을 삭제노드의 자식으로, 오른쪽일 때에도 삭제노드의 자식을 부모의 오른쪽으로 삼는다. 삭제노드가 왼쪽 자식만 가질 때에도 유사한 방법으로 한다. 삭제노드가 두 자식을 모두 가질 때에는 삭제노드의 왼쪽 서브트리에서 가장 큰 노드를 삭제노드 위치로 이동시키고 이동되는 노드의 부모노드와 자식노드를 연결해야한다.

#### - Circular Linked List

: head와 tail이 같은 노드를 가리키는 연결 리스트이다. 노드를 추가할 때 head와 head이전의 노드 사이에 새로운 노드를 추가한다.

#### 4. Result Screen

(↓↗↓↗↓순입니다)

word.txt x

life

인생

job

직업

earth

지구

problem

문제

story

이야기

lot

많은

name

이름

hand

손

place

장소

work

일

make

만들다

different

다른

important

중요한

person

사람

clothes

옷

movie

영화

activity

활동

example

예

letter

편지

fire

불

part

부분

plan

계획

plant

식물

fun

재미

listen

듣다

learn

배우다

each

각각

same

같은

bird

새

trip

여행

vacation

방학

summer

여름

course

과정

command.txt x

ADD

MOVE 10

PRINT TO\_MEMORIZE

UPDATE PROblem 문제

PRINT MEMORIZING I\_IN

UPDATE Story 썰

PRINT MEMORIZING R\_IN

PRINT MEMORIZING I\_PRE

PRINT MEMORIZING R\_PRE

PRINT MEMORIZING R\_POST

PRINT MEMORIZING I\_POST

SEARCH Letter

SEARCH JoB

TEST WORK 일

TEST HanD 손

PRINT MEMORIZED

SAVE

EXIT

```

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

===== MOVE =====
Success
=====

===== PRINT =====
make 만들다
different 다른
important 중요한
person 사람
clothes 옷
movie 영화
activity 활동
example 예
letter 편지
fire 불
part 부분
plan 계획
plant 식물
fun 재미
listen 듣다
learn 배우다
each 각각
same 같은
bird 새
trip 여행
vacation 방학
summer 여름

```

```

lot 많은
name 이름
place 장소
problem 문제
story 썰
work 일
=====

===== PRINT =====
problem 문제
place 장소
hand 손
earth 지구
life 인생
lot 많은
job 직업
name 이름
story 썰
work 일
=====

===== PRINT =====
problem 문제
place 장소
hand 손
earth 지구
life 인생
lot 많은
job 직업
name 이름
story 썰

```

```

===== ERROR =====
600
=====

===== SEARCH =====
Job 직업
=====

===== TEST =====
Pass
=====

===== TEST =====
Pass
=====

===== PRINT =====
work 일
hand 손
=====

===== SAVE =====
Success
=====

youngseo@ubuntu:~/ds$

```

```

course 과정
=====

===== UPDATE =====
Problem 문제 -> 문제
=====

===== PRINT =====
earth 지구
hand 손
job 직업
life 인생
lot 많은
name 이름
place 장소
problem 문제
story 이야기
work 일
=====

===== UPDATE =====
Story 이야기 -> 썰
=====

===== PRINT =====
earth 지구
hand 손
job 직업
life 인생

```

```

work 일
=====

===== PRINT =====
earth 지구
job 직업
name 이름
lot 많은
life 인생
hand 손
story 썰
work 일
place 장소
problem 문제
=====

===== PRINT =====
earth 지구
job 직업
name 이름
lot 많은
life 인생
hand 손
story 썰
work 일
place 장소
problem 문제
=====

```

log.txt x

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

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

```
===== PRINT =====  
make 만들다  
different 다른  
important 중요한  
person 사람  
clothes 옷  
movie 영화  
activity 활동  
example 예  
letter 편지  
fire 불  
part 부분  
plan 계획  
plant 식물  
fun 재미  
listen 듣다  
learn 배우다
```

```
each 각각  
same 같은  
bird 새  
trip 여행  
vacation 방학  
summer 여름  
course 과정
```

```
=====  
===== UPDATE =====  
PROblem 문제 -> 문제  
=====
```

```
===== PRINT =====  
earth 지구  
hand 손  
job 직업  
life 인생  
lot 많은  
name 이름  
place 장소  
problem 문제  
story 이야기  
work 일
```

```
=====  
===== UPDATE =====  
story 이야기 -> 썰  
=====
```

```
===== PRINT =====  
earth 지구  
hand 손  
job 직업  
life 인생  
lot 많은  
name 이름  
place 장소  
problem 문제  
story 썰  
work 일  
=====
```

```
===== PRINT =====  
problem 문제  
place 장소  
hand 손  
earth 지구  
life 인생
```

```
lot 많은  
job 직업  
name 이름  
story 썰  
work 일  
=====
```

```
===== PRINT =====  
problem 문제  
place 장소  
hand 손  
earth 지구  
life 인생  
lot 많은  
job 직업  
name 이름  
story 썰  
work 일  
=====
```

```
===== PRINT =====  
earth 지구  
job 직업  
name 이름
```

```
lot 많은  
life 인생  
hand 손  
story 썰  
work 일  
place 장소  
problem 문제  
=====
```

```
===== PRINT =====  
earth 지구  
job 직업  
name 이름  
lot 많은  
life 인생  
hand 손  
story 썰  
work 일  
place 장소  
problem 문제  
=====
```

```
===== ERROR =====  
600  
=====
```

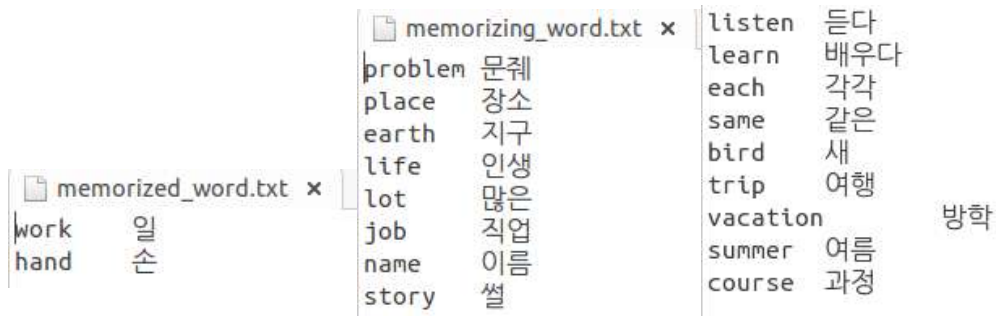
```
===== SEARCH =====  
Job 직업  
=====
```

```
===== TEST =====  
Pass  
=====
```

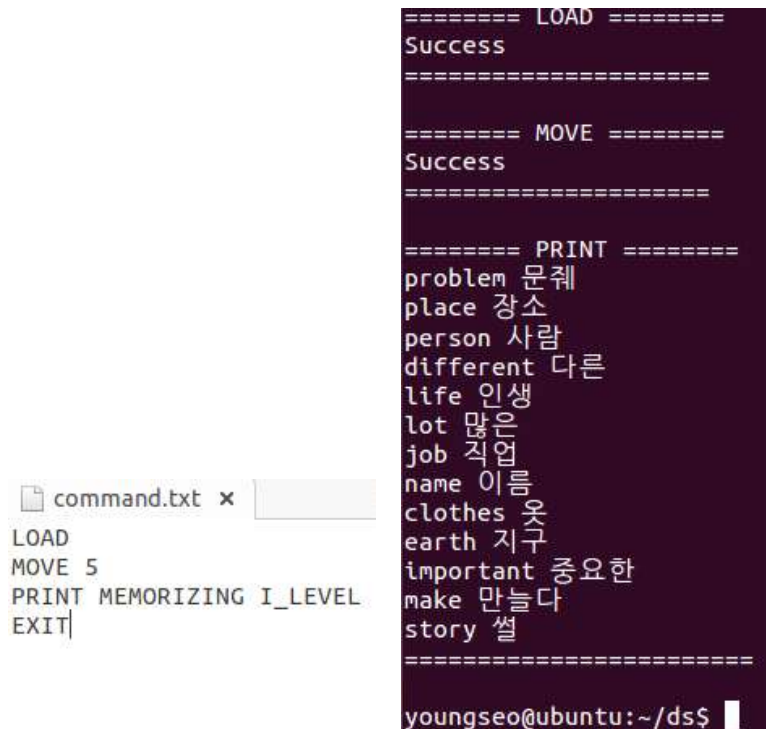
```
===== TEST =====  
Pass  
=====
```

```
===== PRINT =====  
work 일  
hand 손
```

```
===== SAVE =====  
Success  
=====
```



위 파일을 이용



## 5. Consideration

이름	프로젝트에서 맡은 역할	본인 스스로 생각하는 자신의 점수(10)
이영서	전반적인 코드 작성 및 레포트 작성	10
고찰	<p>프로그램 구현할 때 파일을 입력받을 때 한글이 깨져서 발생하는 문제가 있었는데, c++에서는 인코딩을 ansi로, 리눅스에서는 utf-8로 설정하지 않아 발생한 문제였습니다. 개인적으로는 delete가 포함된 test메뉴와 반복문을 이용한 post-order를 이해하고 구현하는 것이 가장 어려운 일이었습니다.</p> <p>팀장은 아니었지만 나아서 팀원별로 분야를 나누어 진행하기로 하였는데, 팀원들이 협조하지 않았습니다. 팀 활동이기에 팀원들과 함께 했다면 좀 더 수월했을 것 같은데 그 점이 아쉽습니다.</p>	