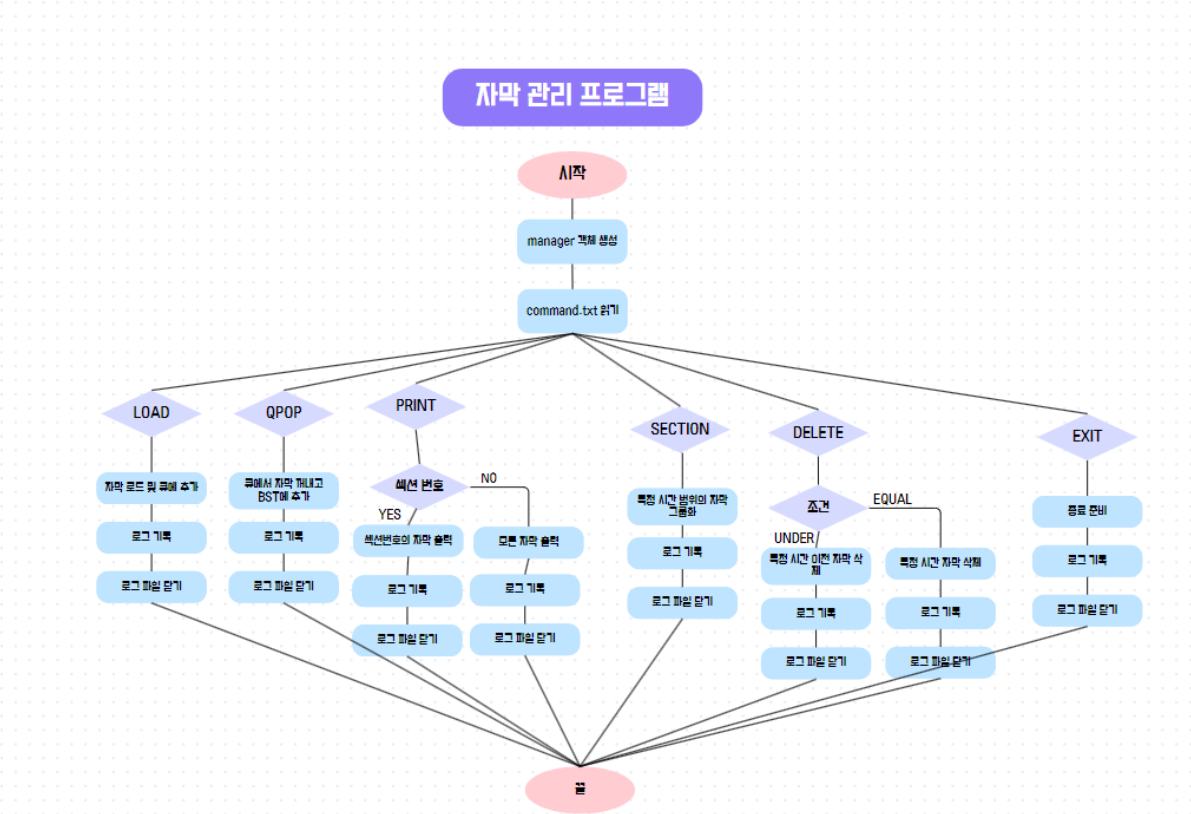
프로젝트 보고서

2023202059 김나희
실습:목
설계:금

Introduction

이 프로젝트는 자막 관리 시스템을 설계하고 구현하는 것을 목표로 한다. 자막 데이터는 큐(Queue)와 이진 탐색 트리(Binary Search Tree, BST), 연결리스트(Linked List) 자료 구조를 이용하여 관리한다. 프로그램은 자막 시간과 내용을 효율적으로 저장한다. LOAD 명령어를 실행하여 자막 데이터 파일로부터 자막 시간과 내용을 읽어 Subtitle_Queue에 자장한다. Queue는 FIFO방식으로 동작하여, 가장 먼저 입력된 자막이 가장 먼저 처리된다. Queue가 구축된 후 Qpop 명령어를 실행하면, Subtitle_Queue에서 데이터를 하나씩 꺼내어 Subtitle_BST에 저장한다. BST는 자막 시간과 자막 내용을 갖는 노드로 구성되며 자막 시간을 기준으로 자막 노드를 정렬하여, 특정 시간 구간의 자막을 빠르게 검색할 수 있도록 돕는다. SECTION 명령어를 실행하여 사용자가 지정한 시간 범위에 해당하는 자막을 Subtitle_BST에서 검색하고, 검색된 자막은 Section_List라는 연결 리스트 구조에 저장한다. 이 리스트는 헤더 노드와 내용 노드로 구성되며, 헤더 노드는 섹션번호, 내용 노드 포인터 정보를 가지며, 내용 노드는 자막시간, 자막내용 정보를 갖고 자막들을 효율적으로 관리하여 출력하거나, 이후 명령어로 처리할 수 있게 한다.

Flowchart

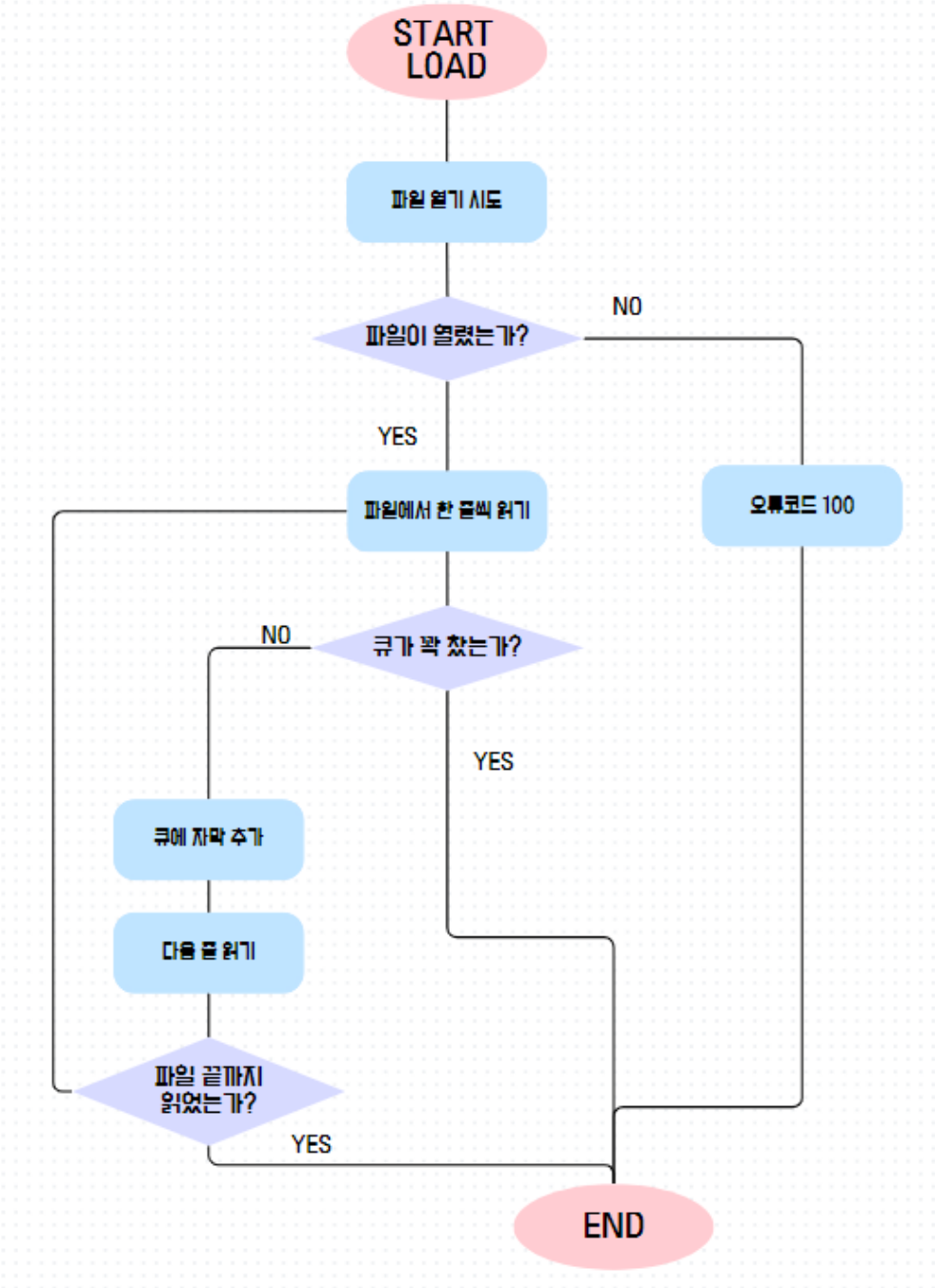


<자막 관리 프로그램의 전체적인 플로우차트>

명령어 파일을 열고 한 줄씩 읽기: 파일에서 명령어를 한 줄씩 읽고, 해당 명령어를 처리하기 위해 분기한다.

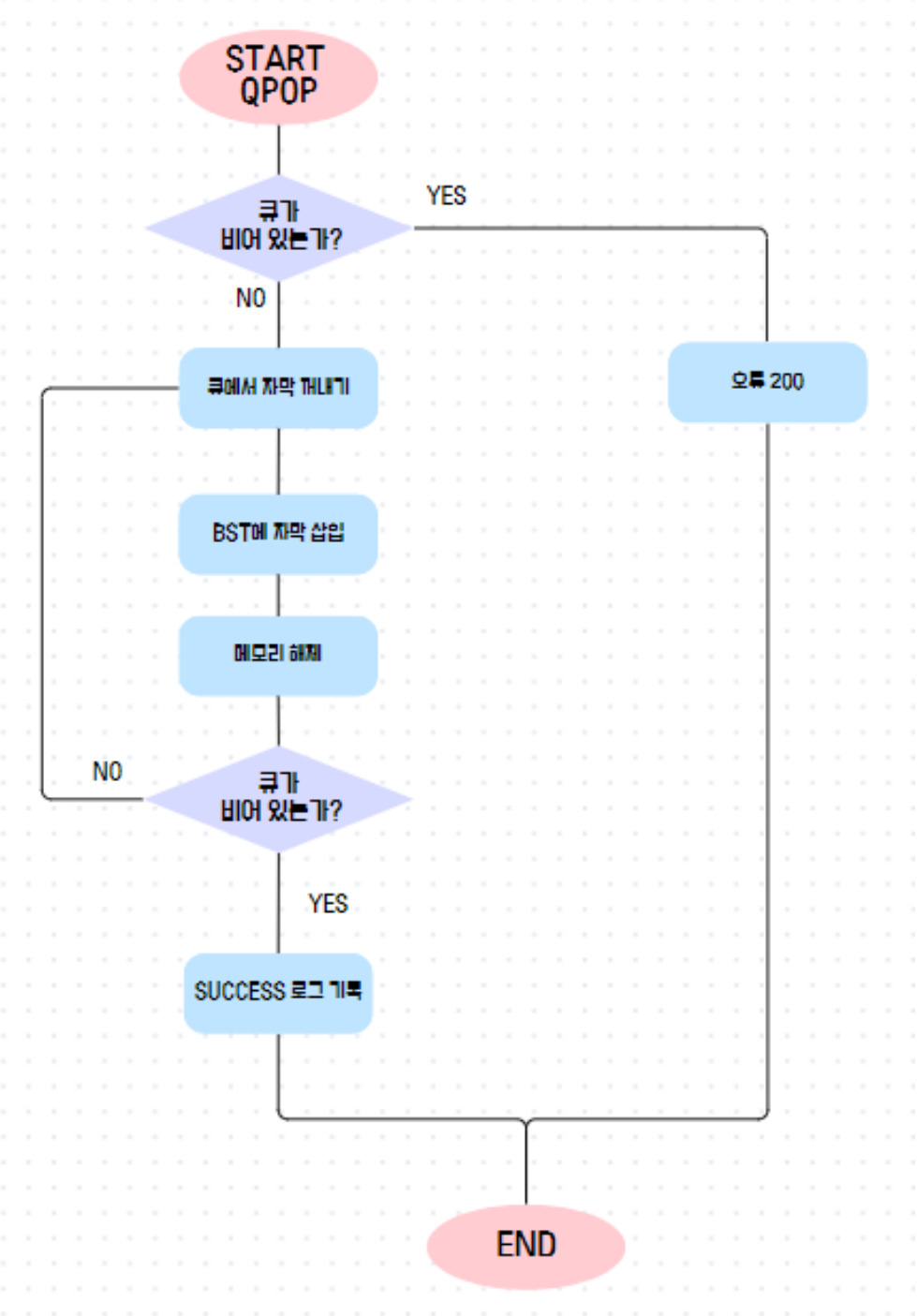
각 명령어 실행: 명령어에 따라 처리 내용이 다르며, 각 처리 후 로그에 기록한다.

EXIT 명령어: 프로그램 종료를 담당하며, 명령어 처리 후 모든 과정을 끝낸다.



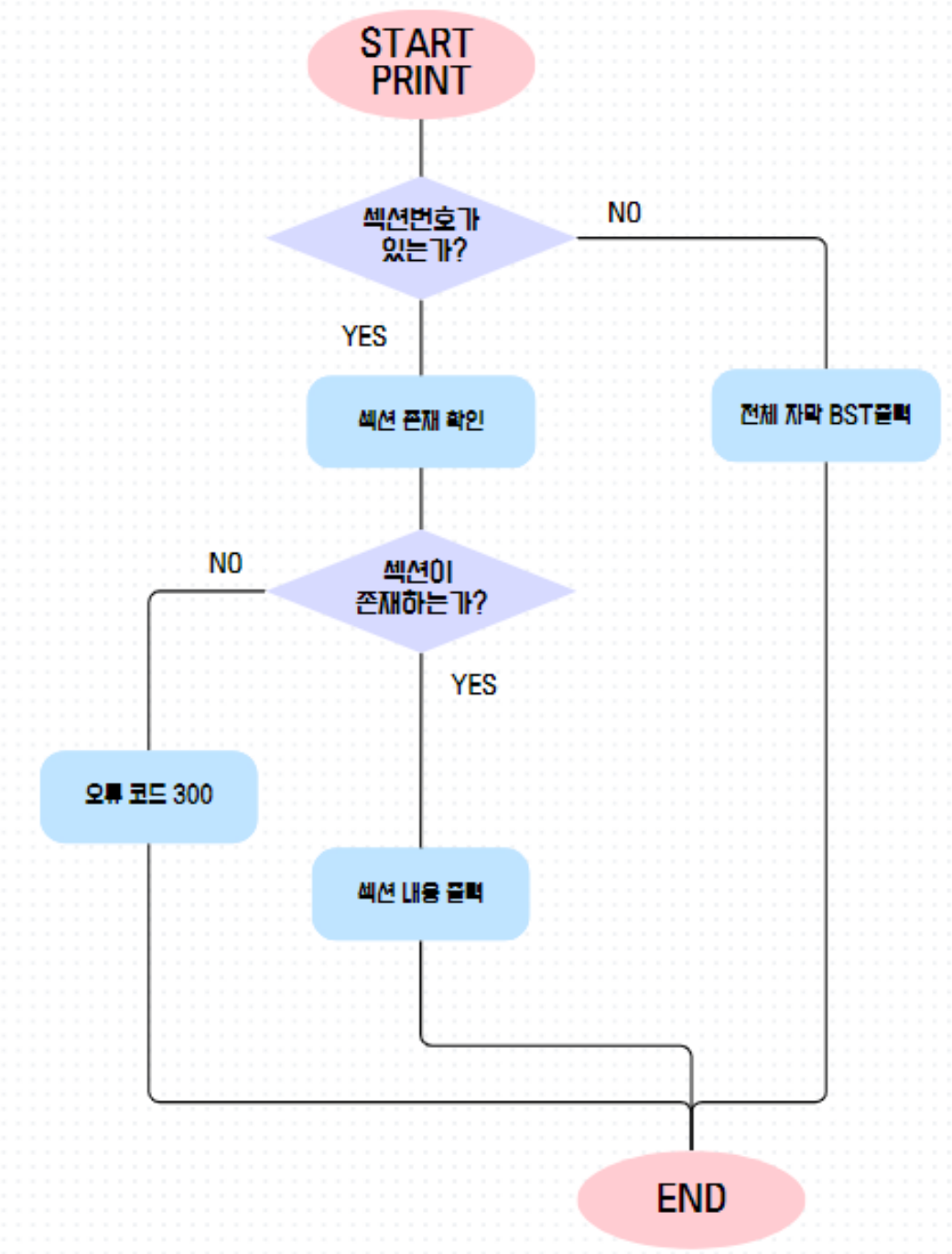
<LOAD 명령어>

LOAD 명령어의 플로우 차트는 자막 파일을 불러와 큐에 데이터를 저장하는 과정을 시각적으로 표현한다. 주된 흐름은 자막 파일을 읽고 큐에 자막을 추가하며, 파일을 열 수 없거나 큐가 꽉 차 있으면 오류를 기록하는 방식이다.



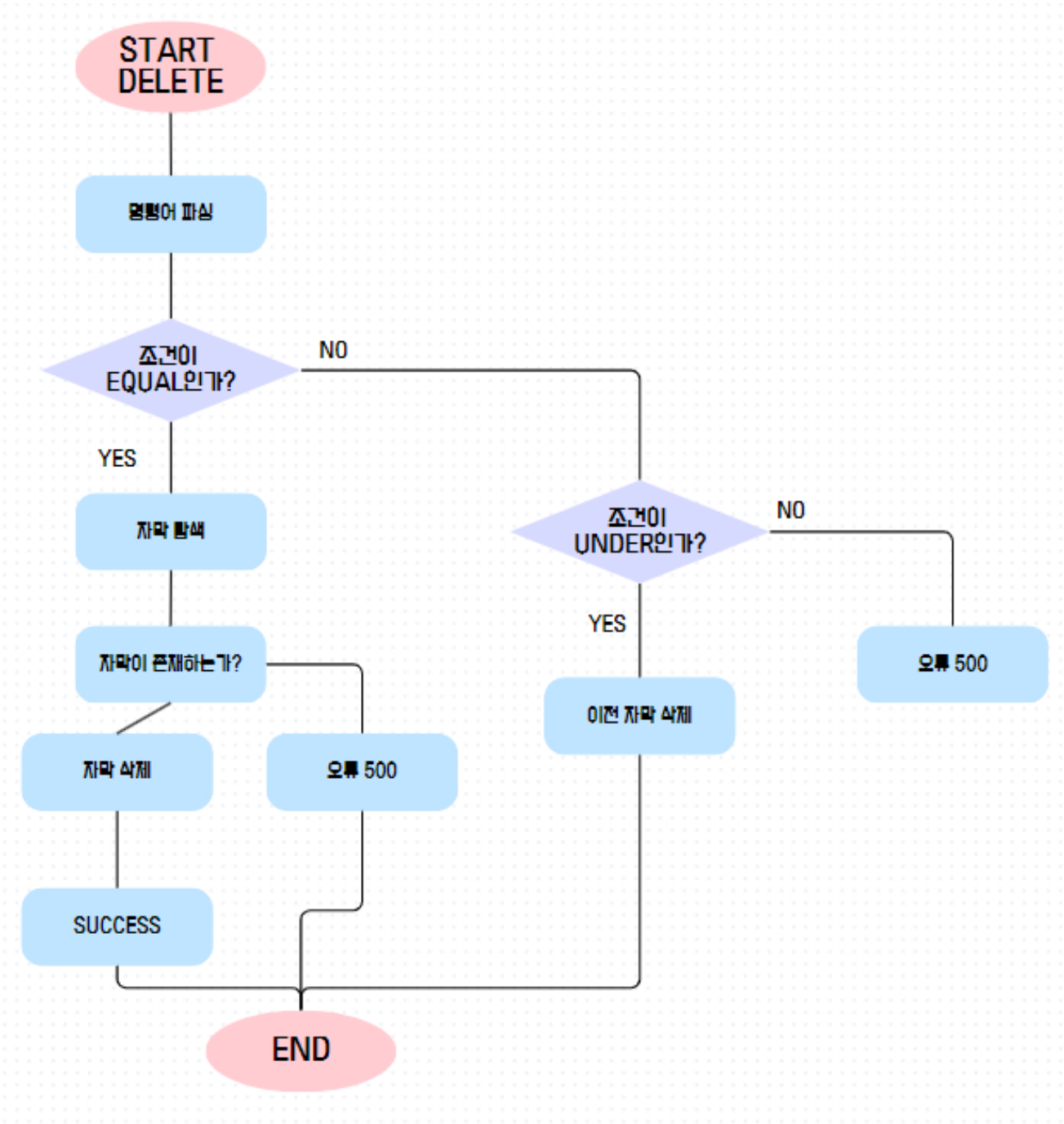
<QPOP 명령어>

QPOP 플로우차트는 큐에서 자막 정보를 꺼내어 BST에 정렬하는 과정을 명확하게 시각화하여, 프로그램의 데이터 흐름과 구조를 이해하는 데 도움을 준다. 이 과정을 통해 자막 정보를 효율적으로 관리하고 검색할 수 있도록 하는 것이 QPOP의 주요 목적이다.



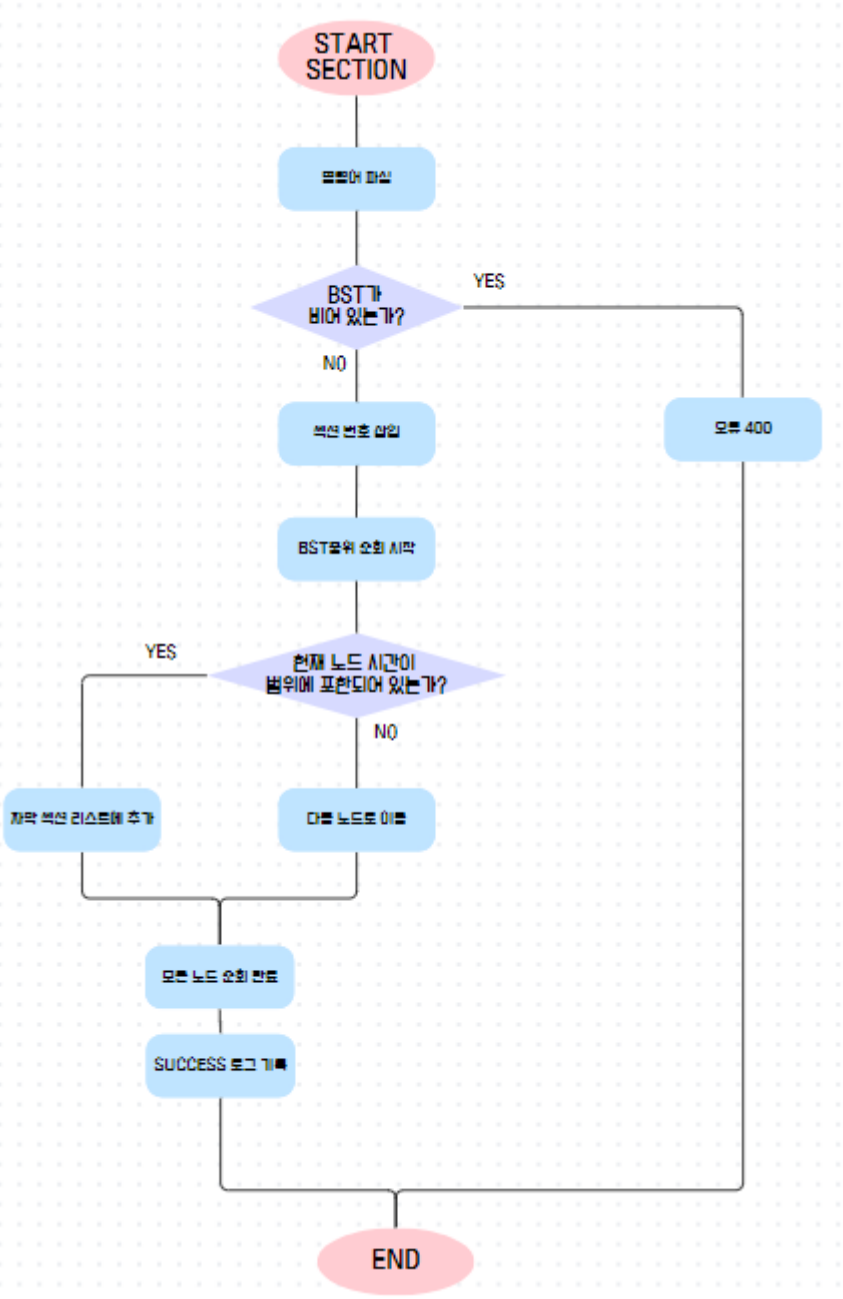
<PIRNT 명령어>

PRINT 명령어의 플로우 차트에서는 섹션 번호가 주어진 경우와 그렇지 않은 경우로 나뉘어 처리된다. 이를 명확하게 플로우 차트로 표현하는 방법은 조건 분기와 그에 따른 다른 흐름을 시각화하는 것이다.



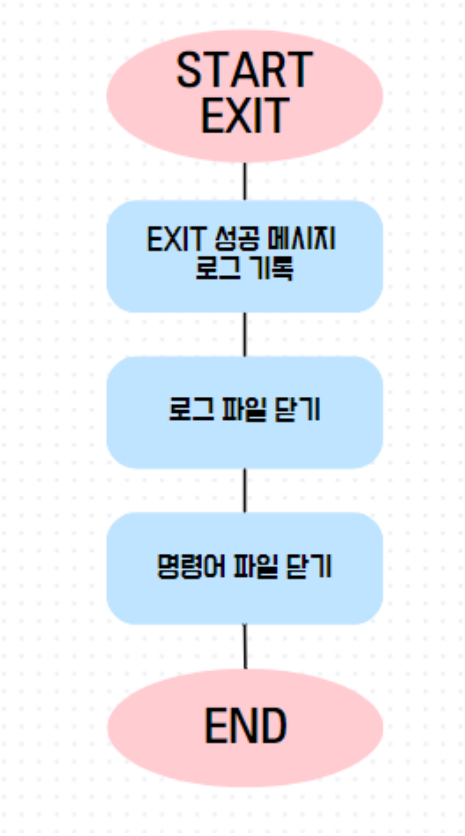
<DELETE 명령어>

DELETE 명령어의 플로우 차트는 "시간(time)" 조건에 따라 자막을 삭제하는 과정을 보여준다. 이 명령어는 "EQUAL" 또는 "UNDER" 조건에 따라 자막을 삭제하므로, 조건에 따른 분기 처리를 포함해야 한다.



<SECTION 명령어>

SECTION 명령어의 플로우 차트는 주어진 시간 범위에 따라 자막을 특정 섹션으로 그룹화하는 과정을 보여준다. 이 명령어는 BST(Binary Search Tree)를 중위 순회하면서 자막을 검색하고, 섹션 리스트에 자막을 추가하는 과정으로 나뉜다.



<EXIT 명령어>

EXIT 명령어는 프로그램 실행을 종료합니다.

Algorithm

이 프로젝트는 Queue, Binary Search Tree(BST), Linked List와 같은 여러 데이터 구조를 사용하여 자막 관리 및 검색을 효율적으로 수행한다.

1. 자막 로딩 및 큐 관리

<LOAD 명령어 처리>

목적: 자막 파일에서 자막 정보를 읽고 이를 큐에 저장하여 후속 처리 및 검색이 용이하도록 한다.

작동과정:

1. load메서드를 호출하여 subtitle.txt파일을 연다.
2. 파일을 한 줄씩 읽어들여 각 줄에서 자막 시간을 추출하고 나머지 내용을 자막 내용으로 설정한다.
3. 큐가 가득 차지 않았을 경우, subtitlequeueNode를 생성하여 큐에 추가 한다.
4. 추가된 자막 정보를 로그 파일에 기록한다.

이 알고리즘은 자막 정보를 효과적으로 관리하기 위해 큐를 사용하며, 큐는 FIFO구

조를 가지므로 데이터가 삽입된 순서대로 소비되어야 하는 자막의 특성과 잘 맞는다.

2. 큐에서 자막 팝핑

<QPOP 명령어 처리>

목적: 큐에서 자막을 꺼내어 BST에 삽입함으로써 자막 정보를 정렬한다.

작동과정:

1. 큐 상태 확인 : qpop메서드를 호출하여 큐가 비어 있는지 확인한다. 큐가 비어 있는 경우 오류코드를 로그에 기록한다.
2. 큐에서 자막 팝핑 : 큐가 비어 있지 않다면 반복문을 사용하여 큐에서 자막 노드를 꺼낸다. 각 반복에서 queue.pop메서드를 호출하여 큐의 가장 앞쪽 노드를 가져온다. 이 노드는 시간과 텍스트 정보를 포함하고 있다. 꺼낸 노드를 BST에 삽입하기 위해 bst.insert메서드를 호출한다. 이진 탐색 트리의 삽입 과정은 노드의 시간 값을 기준으로 적절한 위치에 새로운 노드를 배치한다.
3. 메모리 관리: 삽입이 완료된 후, 꺼낸 노드에 대한 메모리를 해제하여 메모리 누수를 방지한다. 이는 프로그램의 안정성을 높이고, 시스템 자원을 효율적으로 관리하는데 중요하다.

이 알고리즘은 큐에서 자막을 꺼내어 BST에 정렬하여 효율적인 검색과 관리를 가능하게 한다. BST는 이진 탐색 트리의 특성을 가져, 자막 정보를 시간에 따라 효율적으로 검색할 수 있다.

3. 섹션 정의 및 관리

<SECTION 명령어 처리>

목적: 특정 시간 범위에 따라 자막을 섹션으로 그룹화한다.

작동과정:

1. 인자 입력: section메서드를 호출하여 주어진 섹션 번호와 시작/종료 시간을 입력 받는다. 이 단계에서 입력이 올바른지 검증하고 올바르지 않으면 오류코드를 기록한다.
2. 새 섹션 생성 : 섹션 리스트에 새로운 섹션을 추가한다. sectionList.insertSection을 호출하여 새로운 섹션 해더 노드를 생성하고 리스트에 삽입한다. 이 섹션은 이후 자막들을 그룹화 하는 역할을 한다.
3. BST 탐색 및 자막 추가 : BST의 루트 노드에서 시작하여 중위 순회를 통해 모든 노드를 방분한다. 이 과정은 inOrderSectionAdd 메서드를 통해 수행한다. 각 노드를 방문 할 때마다 노드의 시간이 주어진 시작 및 종료 시간 범위에 포함되는지 확인한다. 만약 노드의 시간이 범위에 포함된다면, 해당 자막

을 새로운 섹션에 추가한다. sectionList.addSubtitleToSection(sectionNumber, new SubtitleListNode(node->getTime(), node->getText()))메서드를 호출하여 섹션 리스트에 자막 노드를 추가한다. 시간 범위 미포함이면 무시하고 다음 노드로 이동한다..

4. 새로우 섹션이 생성될 때, 해당 섹션 번호와 자막의 연결을 관리한다.

이 알고리즘은 BST를 중위 순회하면서 자막의 시간 값을 비교하여 특정 범위에 있는 자막을 효과적으로 그룹화한다. 이를 통해 사용자는 원하는 시간 범위의 자막을 쉽게 조회할 수 있다.

4. 자막 출력 및 섹션 출력

<PRIN 명령어 처리>

목적: 현재 BST에 저장된 모든 자막을 출력하거나, 특정 섹션의 자막을 출력한다.

작동과정:

1. 명령어 수신: 사용자가 "PRINT"명령어를 입력하면, RUN메서드는 이 명령어를 감지하고 처리되는데 PRINT명령어는 두 가지 형태로 처리된다.
   - 모든 자막 출력 : PRINT만 입력된 경우, BST에 저장된 모든 자막을 출력한다.
   - 특정 섹션 출력 : PRINT뒤에 섹션 번호가 입력된 경우, 해당 섹션에 포함된 자막만 출력한다.
2. 모든 자막 출력 : print 메서도가 호출되어 BST에서 자막을 출력하는 과정이 시작된다. 로그 파일에는 Subtitle_BST라는 제목이 추가되어 출력된 자막이 BST에서 가져온 것임을 쉽게 인식할 수 있게 한다. bst.printInOrder(flog)메서드를 호출하여 BST의 노드를 중위 순회한다. 중위 순회 방식은 BST의 성질에 따라 왼쪽 서브트리, 현재 노드, 오른쪽 서브트리 순으로 노드를 방문하여, 이 과정에서 각 노드의 자막 정보를 로그 파일에 기록한다.
3. 특정 섹션 출력 : printsection메서드가 호출된다. 이 메서드는 사용자가 요청한 특정 섹션 번호를 기준으로 섹션 리스트에서 자막을 찾고 출력한다. 먼저 searchSection메서드를 통해 해당 섹션이 섹션 리스트에 존재하는지 확인한다. 존재하지 않는 경우, 오류 메시지를 로그 파일에 기록. 섹션이 유효한 경우, 로그파일에 섹션번호를 출력한다. 섹션에 포함된 자막 노드를 가져오기 위해 getContetnHead 메서드를 호출하여 섹션의 첫 번째 자막 노드 포인터를 얻고 자막 노드를 순회하며 각 노드의 정보를 로그 파일에 출력하고 포인터가 nullptr에 도달할때까지 반복

이 알고리즘은 BST의 중위 순회 방식을 사용하여 자막을 정렬된 순서로 출력하

며, 특정 섹션의 자막을 쉽게 찾아볼 수 있도록 한다.

5. 자막 삭제

<DELETE 명령어 처리>

목적: 사용자가 요청한 자막을 삭제하여 자막 목록에서 제거한다. 삭제는 특정 시간의 자막 또는 특정 조건을 만족하는 자막을 대상으로 할 수 있다.

작동과정:

1. 명령어 수신 : 사용자가 "DELETE"명령어를 입력하면, RUN메서드는 이 명령어를 감지하고 처리되는데 DELETE명령어는 두 가지 형태로 처리된다.
   - 특정 시간의 자막 삭제 : DELETE EQUAL <시간> 형식으로 입력된 경우, 해당 시간에 해당하는 자막을 삭제한다.
   - 특정 시간 이전의 자막 삭제 : DELELTE UNDER <시간> 형식으로 입력된 경우, 해당 시간 이전의 모든 자막을 삭제한다.
2. 특정 시간의 자막 삭제 : 사용자가 DELETE EQUAL <시간>을 입력하면 deleteNode메서드가 호출된다. 메서드는 BST에서 특저 시간에 해당하는 자막을 찾기 위해 bst.search메서드를 호출한다. 만약 해당 시간에 자막이 BST에 존재하면 bst.deleteNode메서드를 호출하여 자막을 삭제하다. 삭제가 성공적으로 이루어지면 로그 파일에 성공 메시지를 기록한다. 만약 삭제할 자막이 존재하지 않으면 오류 코드를 기록한다.
3. 특정 시간 이전의 자막 삭제:
   사용자가 DELELTE UNDER <시간>을 입력하면, 동일한 deleteNode메서드가 호출되지만 condition 매개변수는 under로 설정된다. 메서드는 BST에서 지정된 시간보다 이전의 모든 자막을 삭제하기 위해 bst.deleteUnder메서드를 호출한다. 삭제 작업이 완료되면, 성공 메시지를 로그 파일에 기록하고, 만약 BST가 비어 있다면 오류코드를 기록한다.

이 알고리즘은 BST의 검색 및 삭제 기능을 활용하여 자막 정보를 효율적으로 관리한다. 삭제 조건에 따라 적절한 노드를 찾아 삭제함으로써 시스템의 유연성을 높이고, 필요 없는 자막을 효과적으로 제거한다.

자막 관리 시스템은 Queue, Binary Search Tree, Linked List를 조합하여 자막 정보를 효율적으로 관리하고, 다양한 명령어를 통해 사용자가 원하는 정보를 쉽게 처리할 수 있도록 설계되었다. 각 알고리즘은 명확한 목적을 가지고 있으며, 이로 인해 시스템의 성능과 사용 편의성을 향상시킨다. 향후 개선 사항으로는 사용자 인터페이스의 개선, 오류 처리의 강화, 데이터 저장 방식의 최적화 등을 고려할 수 있다.

Result Screen

LOAD

자막 데이터 파일을 읽어와 Subtitle_Queue에 데이터를 저장하는 작업을 수행한다.
자막 시간과 자막 내용을 읽어서 Queue에 순서대로 삽입한다. 자막 파일을 성공적으로 읽어 Queue에 삽입된 자막 정보들이 출력되고 자막 시간 – 자막 내용이 표시된다.

```
==== LOAD ====
01:03:45 -  You're not listening to me!
00:52:36 -  You're welcome.
00:54:16 -  Hey, you! Where's the other mother?
00:54:18 -  I wanna go home.
00:53:50 -  Mom! Dad!
00:53:56 -  Oh, God. I'm still here?
00:54:19 -  All will be swell, soon as Mother's refreshed.
00:54:23 -  Her strength is our strength.
00:52:21 -  Bed? Before dinner?
00:52:20 -  Right now!
00:52:19 -  I'm going to bed.
00:21:37 -  See you soon.
00:52:23 -  I'm really, really tired. Yeah.
01:09:34 -  Ok.
01:09:24 -  Challenge her, then.
01:09:29 -  She's got a thing for games.
01:09:26 -  She may not play fair, but she won't refuse.
01:09:17 -  I have to go back. They are my parents.
01:11:36 -  A finding things game.
01:11:27 -  Everybody likes games.
01:11:51 -  What if you don't find them?
01:11:59 -  And I'll let you sew buttons into my eyes.
01:11:54 -  If I lose, I'll stay here with you forever
01:11:31 -  What kind of game would it be?
01:11:38 -  And what is it you'd be finding?
01:11:41 -  My real parents. And... the eyes of the children.
01:11:26 -  I know you like them.
01:11:22 -  Why don't we play a game?
===============
```

Qpop

Queue에서 데이터를 하나씩 꺼내 Subtitel_BST에 삽입하는 역할을 한다.
Queue에서 자막을 하나씩 꺼내 이진 탐색 트리에 저장하고 각 자막 노드들은 자막 시간이 기준이 되어 정렬된다. 자막 데이터가 Queue에서 성공적으로 꺼내어지면 아래와 같은 출력이 된다.

```
===== QPOP =====
Success
===============
```

Print

Subtitle_BST에 저장된 모든 자막을 시간 순서대로 출력하는 역할을 한다.

이진 탐색 트리에 저장된 모든 자막 데이터를 중위 순회를 통해 시간 순으로 출력한다. 자막 시간에 따라 오름차순으로 정렬된 상태로 출력되며, 각각의 자막 시간과 자막 내용이 출력된다.

```
===== PRINT =====
Subtitle_BST
00:21:37 -  See you soon.
00:52:19 -  I'm going to bed.
00:52:20 -  Right now!
00:52:21 -  Bed? Before dinner?
00:52:23 -  I'm really, really tired. Yeah.
00:52:36 -  You're welcome.
00:53:50 -  Mom! Dad!
00:53:56 -  Oh, God. I'm still here?
00:54:16 -  Hey, you! Where's the other mother?
00:54:18 -  I wanna go home.
00:54:19 -  All will be swell, soon as Mother's refreshed.
00:54:23 -  Her strength is our strength.
01:03:45 -  You're not listening to me!
01:09:17 -  I have to go back. They are my parents.
01:09:24 -  Challenge her, then.
01:09:26 -  She may not play fair, but she won't refuse.
01:09:29 -  She's got a thing for games.
01:09:34 -  Ok.
01:11:22 -  Why don't we play a game?
01:11:26 -  I know you like them.
01:11:27 -  Everybody likes games.
01:11:31 -  What kind of game would it be?
01:11:36 -  A finding things game.
01:11:38 -  And what is it you'd be finding?
01:11:41 -  My real parents. And... the eyes of the children.
01:11:51 -  What if you don't find them?
01:11:54 -  If I lose, I'll stay here with you forever
01:11:59 -  And I'll let you sew buttons into my eyes.
===============
```

Print 3(section number)

Section_List에 저장된 자막을 섹션 번호별로 시간 순서에 맞게 출력한다. 출력은 섹션 번호와 해당 섹션에 포함된 자막의 시간과 자막 내용이 출력된다.

```
===== PRINT =====
Section 3
01:03:45 -  You're not listening to me!
01:09:17 -  I have to go back. They are my parents.
01:09:24 -  Challenge her, then.
01:09:26 -  She may not play fair, but she won't refuse.
01:09:29 -  She's got a thing for games.
01:09:34 -  Ok.
===============
```

SECTION 3 01:00:00 01:10:00

사용자가 지정한 시간 범위에 해당하는 자막을 Subtitle_BST에서 탐색하고, 그 결과를 Section_List에 저장한다. BST에서 지정한 시간 범위에 해당하는 자막을 탐색하고, 해당 자막들을 Section_List로 구성한다.

```
===== SECTION =====
Success
===============
```

DELETE EQUAL 01:09:17

Subtitle_BST에서 지정된 자막 시간을 가진 노드를 찾아 삭제하는 역할을 한다. 이 명령어를 사용하여 특정 시간의 자막을 제거할 수 있다. Subtitle_BST에서 지정된 자막 시간을 가진 노드를 탐색한 후, 해당 노드를 삭제한다.

```
===== DELETE =====
Success
===============
```

```
===== PRINT =====
Subtitle_BST
00:21:37 -  See you soon.
00:52:19 -  I'm going to bed.
00:52:20 -  Right now!
00:52:21 -  Bed? Before dinner?
00:52:23 -  I'm really, really tired. Yeah.
00:52:36 -  You're welcome.
00:53:50 -  Mom! Dad!
00:53:56 -  Oh, God. I'm still here?
00:54:16 -  Hey, you! Where's the other mother?
00:54:18 -  I wanna go home.
00:54:19 -  All will be swell, soon as Mother's refreshed.
00:54:23 -  Her strength is our strength.
01:03:45 -  You're not listening to me!
01:09:24 -  Challenge her, then.
01:09:26 -  She may not play fair, but she won't refuse.
01:09:29 -  She's got a thing for games.
01:09:34 -  Ok.
01:11:22 -  Why don't we play a game?
01:11:26 -  I know you like them.
01:11:27 -  Everybody likes games.
01:11:31 -  What kind of game would it be?
01:11:36 -  A finding things game.
01:11:38 -  And what is it you'd be finding?
01:11:41 -  My real parents. And... the eyes of the children.
01:11:51 -  What if you don't find them?
01:11:54 -  If I lose, I'll stay here with you forever
01:11:59 -  And I'll let you sew buttons into my eyes.
===============
```

DELETE UNDER 01:09:17

Subtitle_BST에서 지정된 자막 시간보다 이전의 모든 자막 노드를 삭제하는 역할을 한다. 이를 통해 특정 시간 이전의 자막을 한 번에 제거할 수 있다. 지정된 시간보다 이전에 있는 모든 자막 노드를 탐색하여 삭제한다.

```
===== DELETE =====
Success
===============
```

```
===== PRINT =====
Subtitle_BST
01:09:24 -  Challenge her, then.
01:09:26 -  She may not play fair, but she won't refuse.
01:09:29 -  She's got a thing for games.
01:09:34 -  Ok.
01:11:22 -  Why don't we play a game?
01:11:26 -  I know you like them.
01:11:27 -  Everybody likes games.
01:11:31 -  What kind of game would it be?
01:11:36 -  A finding things game.
01:11:38 -  And what is it you'd be finding?
01:11:41 -  My real parents. And... the eyes of the children.
01:11:51 -  What if you don't find them?
01:11:54 -  If I lose, I'll stay here with you forever
01:11:59 -  And I'll let you sew buttons into my eyes.
===============
```

<01:09:17의 자막은 이미 위에 명령어로 인해 삭제된 상태>

Exit

프로그램을 종료하는 역할을 한다.

```
===== EXIT =====
Success
===============
```

Consideration

이번 프로젝트를 통해 어떤 상황에서 Queue, BST, Linked List 자료 구조를 선택해야 하는지 알게 되었다. Queue는 자막 데이터를 파일에서 읽어오는 순서대로 처리하기에 적합하고 FIFO 방식으로 자막의 순차적 처리가 자연스럽게 이루어지기 때문에 Queue를 사용하였고, BST는 자막 시간이 순서대로 정렬되어 있어야 하고 이를 통해 시간 범위에 따른 자막 검색이 효율적이어서 사용한 것이고, Linked List는 검색된 자막 데이터를 임시로 저장하고 이후 명령어에서 사용활 수 있도록 관리하는데 적합하기 때문에 사용을 했고 이로 인해 다른 프로젝트를 할 때 어떤 자료 구조가 타당한지 알 수 있게 되었고 자료 구조의 중요성을 다시 한 번 깨닫게 되었다. 단순한 자막 관리 시스템에도 각각의 자료 구조가 성능과 유지 보수성에 큰 영향을 미친다는 것을 알게 되었고 또한, 실제 데이터를 처리하는 상황에서 자료 구조의 선택이 얼마나 중요한지 체감할 수 있었다. 추가적으로 수업시간에 배웠던 트리 구조를 활용한 검색 최적화 기법, 그리고 큐와 리스트와같은 선형 구조에서의 효율적인 데이터 처리 방법에 대한 이해가 더욱 깊어졌다. 프로젝트를 진행하면서 많은양의 데이터가 있었다면 시간이 많이 걸릴 것이라고 생각했고 다른 트리의 구조를 사용하면 더 성능을 최적화 할 수 있다고 생각하였다. 프로젝트가 커지면서 코드 관리의 중요성도 깨닫게 되었고 코드 구조를 가독성과 유지 보수성을 높이기 위해

많은 노력을 해야 된다고 생각했다.