데이터구조설계 및 실습

프로젝트#1 보고서

학과: 컴퓨터정보공학부

담당교수: 최상호 교수님

분반: 목요일

학번: 2024402055

성명: 박현지

1. Introduction

본 프로젝트는 이진 탐색 트리(BST), 연결 리스트, 큐 자료 구조를 활용하여 음악 플레이리스트 관리 시스템을 구현하는 것을 목표로 한다. 구현된 시스템은 음악 데이터 파일(Music_List.txt)로부터 가수, 노래 제목, 재생 시간 정보를 읽어 관리하며, 사용자 명령에 따라 음악을 검색, 추가, 삭제하고 플레이리스트를 생성 및 운영한다.

A. 핵심 자료구조

i. ***MusicQueue*** **(Queue) :** 초기 데이터 로딩을 위한 임시 저장소. FIFO 방식으로 데이터를 관리한다.

ii. ***ArtistBST*** **(BST) :** 가수 이름을 기준으로 정렬되어 모든 음악 정보를 저장한다.

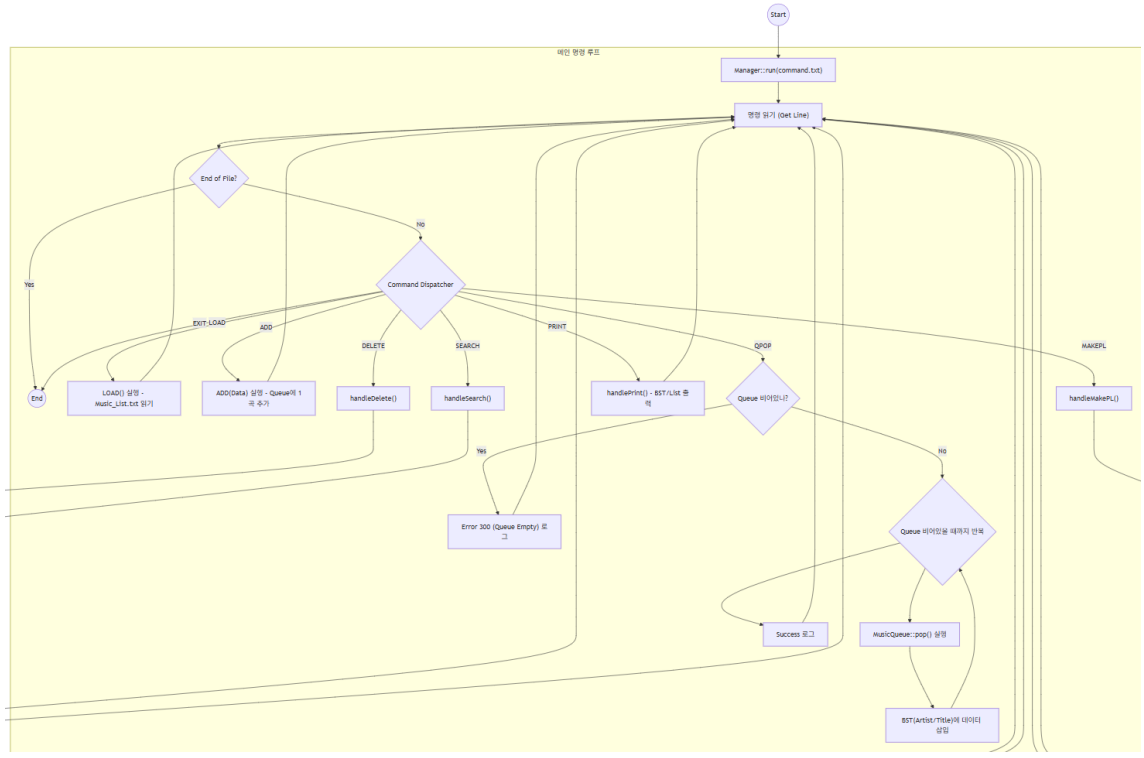
iii. ***TitleBST*** **(BST) :** 노래 제목을 기준으로 정렬되어 모든 음악 정보를 저정한다.

iv. ***PlayList*** **(Circular Linked List) :** 사용자가 선택한 곡들을 담는 실제 플레이리스트. 최대 10곡까지 저장 가능하며, 순환 구조로 구현되어 있다.

Manager 클래스가 전체 시스템을 총괄하며, *command.txt* 파일의 명령 (*LOAD, ADD, QPOP, SEARCH, MAKEPL, PRINT, DELETE, EXIT*)을 파싱하여 각 자료 구조의 기능을 호출하고 결과를 *log.txt*에 기록한다.

2. Flowchart

A. **Main 명령 루프**



i. Start → Manager::run()
프로그램이 시작되면 Manager 객체의 run 함수가 command.txt 파일을 열고 실행을 시작한다.

ii. Manager::run() → 명령 읽기
파일에서 한 줄의 명령을 읽는다.

iii. 명령 읽기 → End of File?
파일의 끝인지 판단한다. Yes면 F((End))로 종료하고, No면 E로 분기한다.

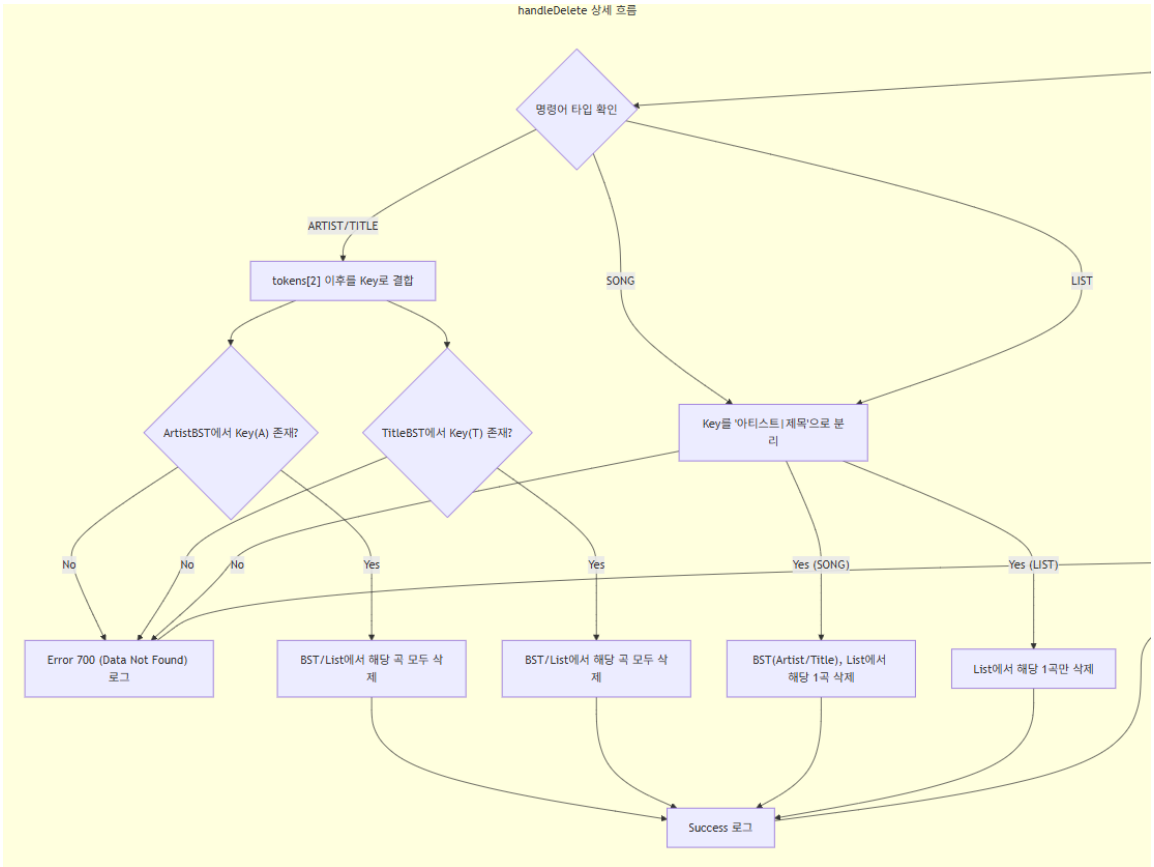
iv. Command Dispatcher
명령의 종류에 따라 H1(LOAD), H2(ADD), I(QPOP), J(SEARCH), K(MAKEPL), L(DELETE), M(PRINT) 또는 F(EXIT)로 분기한다.

v. Queue 비어있나?
큐가 비어있으면 I_Err (Error 300)을 기록하고, 아니면 I_Loop로 진입한다.

vi. Queue 비어있을 때까지 반복
큐가 빌 때까지 I_Pop (MusicQueue::pop)을 반복 실행하며, 꺼낸 데이터를 I_Insert (BST 삽입)로 넘긴다.

vii. Success 로그 → 명령 읽기
QPOP 완료 후 Success 로그를 기록하고 메인 루프로 복귀한다.

B. **handleDelete 상세 흐름**



i. handleSearch() → 명령어 타입 확인
명령이 ARTIST, TITLE, SONG 중 무엇인지 확인한다.

ii. J2[Key로 결합]
다중 단어 키 수정 로직: tokens[2] 이후의 모든 토큰을 하나의 검색 키로 결합한다 (예: "blue moon").

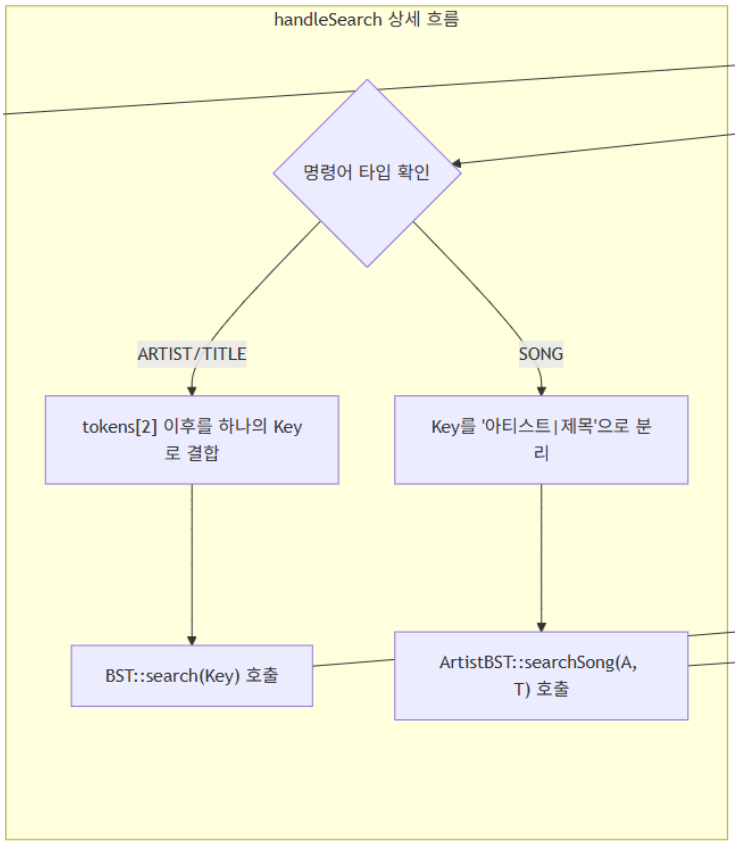
iii. Key 결합 → BST::search(Key) 호출
결합된 키를 해당 BST에 전달하여 검색을 수행한다.

iv. J3[Key를 분리]

v. Key 분리 → ArtistBST::searchSong(A, T)
아티스트 및 제목을 기준으로 곡을 검색한다.

vi. BST 검색 결과 → C
검색 결과를 로그에 기록하고 메인 루프로 복귀한다.

C. **handleSearch 상세 흐름**



i. handleMakePL() → 명령어 타입 확인
명령 타입에 따라 분기한다. ARTIST/TITLE 타입은 K2로 이동하여 키를 결합한다.

ii. Key 결합 → BST::getSongs(Key) 호출
키에 해당하는 모든 곡 정보를 BST에서 가져온다.

iii. 곡 정보 획득 → PlayList 용량(10) 초과?

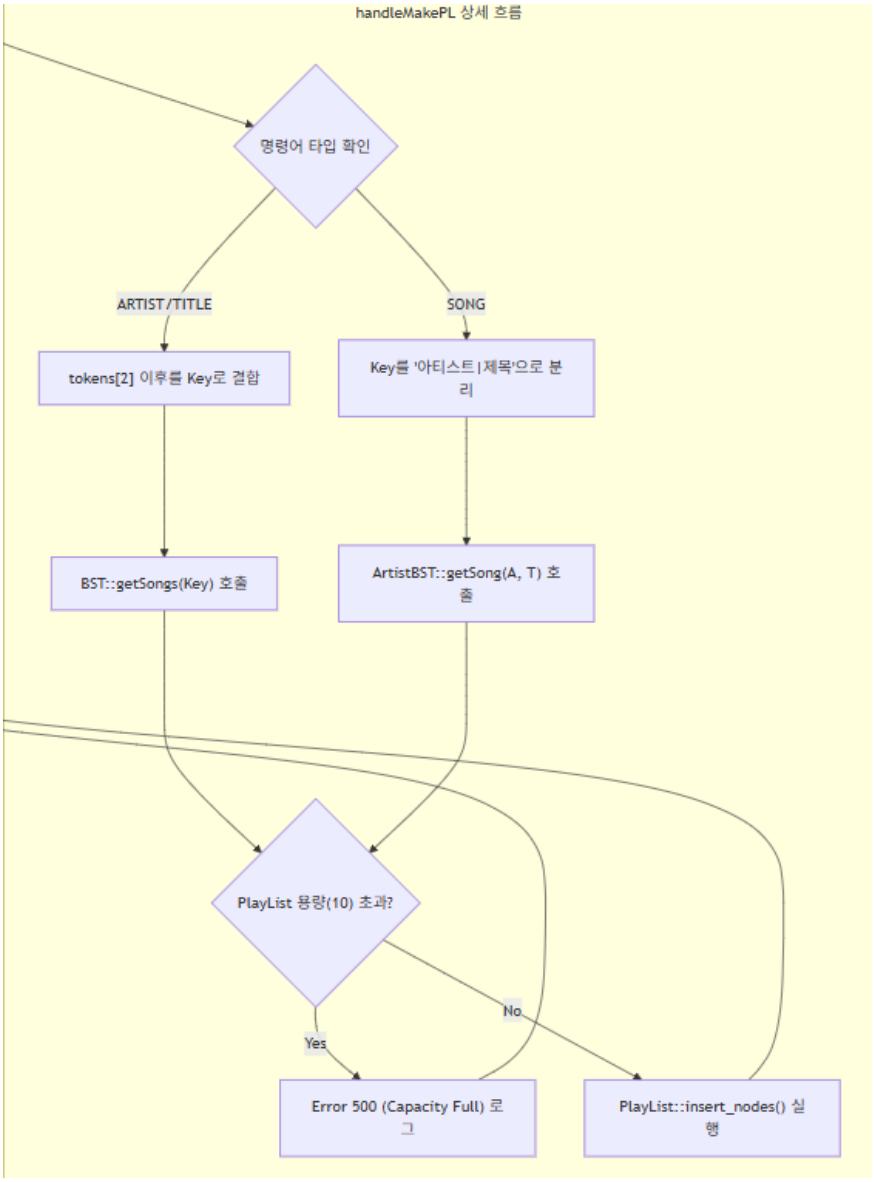
현재 플레이리스트에 곡을 추가할 경우 총 곡 수가 10개를 초과하는지 판단한다.

iv. Yes → Error 500 로그
용량 초과 시 Error 500을 기록한다.

v. No → PlayList::insert_nodes() 실행
용량 초과가 아니면 플레이리스트에 곡들을 추가한다.

vi. 처리 완료 → C
처리 완료 후 메인 루프로 복귀한다.

D. **handleMakePL 상세 흐름**



i. handleDelete() → 명령어 타입 확인
ARTIST, TITLE, SONG, LIST 타입 중 하나를 확인한다.

ii. L2[Key 결합]

다중 단어 키를 하나의 문자열로 결합한다.

iii. Key 결합 → BST 존재 여부 확인
삭제 대상이 BST에 존재하는지 확인한다. 없으면 L_Err (Error 700)으로 분기한다.

iv. Yes → BST/List에서 해당 곡 모두 삭제
해당 키에 연관된 모든 곡을 ArtistBST, TitleBST, PlayList 세 곳에서 모두 삭제하여 데이터 일관성을 유지한다.

v. L3[Key 분리]

vi. Yes (LIST) → List에서 해당 1곡만 삭제
PlayList에서만 해당 곡을 삭제한다.

vii. Error 700 로그 → C
삭제 대상이 없었으면 Error 700을 기록하고 복귀한다.

viii. Success 로그 → C
삭제 성공 로그를 기록하고 메인 루프로 복귀한다.

3. Algorithm

본 프로젝트에 사용된 주요 자료 구조와 알고리즘의 동작 방식은 다음과 같다.

A. 이진 탐색 트리 (BST)의 기본 동작 (*ArtistBST, TitleBST*)

i. **삽입 (Insert)**
QPOP 명령 시 *MusicQueue*에서 데이터를 꺼내 *ArtistBST*와 *TitleBST*에 삽입한다. 노드의 키(아티스트명 또는 노래 제목)를 기준으로 루트 노드부터 비교하며, 작으면 왼쪽, 크면 오른쪽 서브트리로 이동하여 적절한 위치에 새 노드를 추가한다.

ii. **검색 (Search)**
SEARCH 명령 시 사용된다. BST의 탐색 특성을 이용하여 $O(\log N)$의 시간 복잡도로 원하는 키를 가진 노드를 찾아 해당 노드의 모든 곡 정보를 *log.txt*에 기록한다.

1. **다중 키 처리 :** *SEARCH ARTIST [키]* 또는 *SEARCH TITLE [키]* 명령 시, 명령어의 세 번째 토큰부터 끝까지의 모든 문자열을 공백으로 연결하여 완전한 하나의 검색 키로 만들어 정확한 검색을 수행한다.

B. BST의 삭제 및 연관 데이터 처리

i. **삭제 (Delete)**
DELETE 명령 시 호출되며, BST의 표준 삭제 알고리즘을 따른다 (자식 0개, 1개, 2개 케이스 분류).

ii. 데이터 일관성 유지

*DELETE ARTIST [아티스트명]*과 같이 아티스트 단위로 삭제할 경우, *ArtistBST*에서 해당 아티스트의 모든 곡을 삭제한 후, 삭제된 곡 정보를 사용하여 *TitleBST*와 *PlayList*에서도 해당 곡을 찾아 삭제하여 시스템 전체 데이터의 일관성을 유지한다.

C. 플레이리스트 관리 (PlayList)

i. **자료구조 :** 순환 연결 리스트

ii. 삽입 (Insert)

MAKEPL 명령 시 BST에서 검색된 곡들을 플레이리스트에 추가한다.

1. **용량제어 :** 플레이리스트의 최대 용량(10곡)을 초과하는 추가 시도에 대해서는 ***Error 500***을 발생시키고 작업을 취소한다.

iii. 삭제 (Delete)

DELETE LIST [곡 정보] 명령 시 플레이리스트에서 해당 곡만 삭제하며, 연결 리스트의 다음 노드 포인터를 재설정하여 순환 구조를 유지한다.

4. Result Screen

다음은 코드를 실행한 *log.txt*의 주요 내용과 각 명령의 동작을 설명한다.

LOAD & QPOP

```
========LOAD========
N.Flying/blue moon/3:36
N.Flying/moonshot/3:26
N.Flying/songbird/3:46
N.Flying/rooftop/3:19
N.Flying/oh really./3:10
PSY/gangnam style/3:25
PSY/gentleman/3:14
PSY/daddy/3:32
PSY/new face/3:16
PSY/that that/3:05
BTS/dynamite/3:19
IU/blueming/3:37
Michael Jackson/beat it/4:18
Maroon 5/sugar/3:55
Westlife/my love/3:51
Buzz/my love/4:05
Hyorin/blue moon/3:24
====================
========ERROR========
200
====================
========QPOP========
Success
====================
```

BST에 17곡 삽입 완료

SEARCH

```
========SEARCH========
N.Flying/blue moon/3:36
N.Flying/moonshot/3:26
N.Flying/songbird/3:46
N.Flying/rooftop/3:19
N.Flying/oh really./3:10
======================
========SEARCH========
N.Flying/blue moon/3:36
Hyorin/blue moon/3:24
======================
========SEARCH========
Westlife/my love/3:51
Buzz/my love/4:05
======================
========SEARCH========
Error: Song not found
======================
```

ARTIST N.Flying 검색 결과
TITLE blue moon 검색 결과
TITLE my love 검색 결과
목록에 없는 곡 검색(예외처리)

다중 단어 제목 검색 성공

MAKEPL

`` ========MAKEPL======== N.Flying/blue moon/3:36 N.Flying/moonshot/3:26 N.Flying/songbird/3:46 N.Flying/rooftop/3:19 N.Flying/oh really./3:10 Count 5/10 Time 17min 17sec ===================== ========MAKEPL======== N.Flying/blue moon/3:36 N.Flying/moonshot/3:26 N.Flying/songbird/3:46 N.Flying/rooftop/3:19 N.Flying/oh really./3:10 Hyorin/blue moon/3:24 Count 6/10 Time 20min 41sec ===================== ========MAKEPL======== N.Flying/blue moon/3:36 N.Flying/moonshot/3:26 N.Flying/songbird/3:46 N.Flying/rooftop/3:19 N.Flying/oh really./3:10 Hyorin/blue moon/3:24 Westlife/my love/3:51 Buzz/my love/4:05 Count 8/10 Time 28min 37sec ===================== ========ERROR======== 500 ===================== ```		8곡 플레이 리스트에 추가 완료 MAKEPL ARTIST PSY 시도, 플레이 리스트 용량 (8/10) 초과하여 Error 500 정상 출력
PRINT		

========PRINT======== ArtistBST BTS/dynamite/3:19 Buzz/my love/4:05 Hyorin/blue moon/3:24 IU/blueming/3:37 Maroon 5/sugar/3:55 Michael Jackson/beat it/4:18 N.Flying/blue moon/3:36 N.Flying/moonshot/3:26 N.Flying/songbird/3:46 N.Flying/rooftop/3:19 N.Flying/oh really./3:10 PSY/gangnam style/3:25 PSY/gentleman/3:14 PSY/daddy/3:32 PSY/new face/3:16 PSY/that that/3:05 Westlife/my love/3:51 ==================== ========PRINT======== TitleBST Michael Jackson/beat it/4:18 N.Flying/blue moon/3:36 Hyorin/blue moon/3:24 IU/blueming/3:37 PSY/daddy/3:32 BTS/dynamite/3:19 PSY/gangnam style/3:25 PSY/gentleman/3:14 N.Flying/moonshot/3:26 Westlife/my love/3:51 Buzz/my love/4:05 PSY/new face/3:16 N.Flying/oh really./3:10 N.Flying/rooftop/3:19 N.Flying/songbird/3:46 Maroon 5/sugar/3:55 PSY/that that/3:05 ====================		ArtistBST의 중위 순회 결과 출력. TitleBST의 중위 순회 결과 출력. 순환 연결 리스트의 현재 8곡 목록 출력.

========PRINT======== N.Flying/blue moon/3:36 N.Flying/moonshot/3:26 N.Flying/songbird/3:46 N.Flying/rooftop/3:19 N.Flying/oh really./3:10 Hyorin/blue moon/3:24 Westlife/my love/3:51 Buzz/my love/4:05 Count : 8 / 10 Time : 28min 37sec ====================	
DELETE	
========DELETE======== Success ==================== ========ERROR======== 700 ==================== ========DELETE======== Success ==================== ========ERROR======== 700 ==================== ========DELETE======== Success ==================== ========ERROR======== 700 ==================== ========PRINT======== ArtistBST BTS/dynamite/3:19 Buzz/my love/4:05 IU/blueming/3:37 Maroon 5/sugar/3:55 Michael Jackson/beat it/4:18 N.Flying/moonshot/3:26 N.Flying/songbird/3:46 N.Flying/oh really./3:10 Westlife/my love/3:51 ====================	PSY의 5곡을 모든 자료 구조에서 삭제 성공. DELETE ARTIST PSY 재시도 시 존재하지 않는 데이터 삭제에 대한 Error 700 정상 출력. DELETE TITLE blue moon 다중 단어 제목 삭제 성공. DELETE TITLE blue moon 재시도 시 존재하지 않는 데이터 삭제에 대한 Error 700 정상 출력. DELETE TITLE rooftop 모든 자료 구조에서 삭제 성공 재시도 시 700 출력 DELETE TITLE songbird 모든 자료 구조에서 삭제 성공 모든 삭제 작업 후 플레이리스트에 남은 4곡 목록 출력. 종료

```
========DELETE========
Success
======================
========PRINT========
ArtistBST
BTS/dynamite/3:19
Buzz/my love/4:05
IU/blueming/3:37
Maroon 5/sugar/3:55
Michael Jackson/beat it/4:18
N.Flying/moonshot/3:26
N.Flying/oh really./3:10
Westlife/my love/3:51
======================
========PRINT========
TitleBST
Michael Jackson/beat it/4:18
IU/blueming/3:37
BTS/dynamite/3:19
N.Flying/moonshot/3:26
Westlife/my love/3:51
Buzz/my love/4:05
N.Flying/oh really./3:10
Maroon 5/sugar/3:55
======================
========PRINT========
N.Flying/moonshot/3:26
N.Flying/oh really./3:10
Westlife/my love/3:51
Buzz/my love/4:05
Count : 4 / 10
Time : 14min 32sec
======================
=======EXIT========
Success
====================
```

5. Consideration

본 프로젝트를 통해 복잡한 데이터 관리 시스템에서 여러 자료 구조의 상호작용과 데이터 일관성 유지의 중요성을 깊이 이해했다.

A. 해결한 문제

i. 명령어 파싱 개선

*SEARCH*와 *DELETE* 명령에서 다중 단어로 구성된 키(예: "blue moon", "gangnam style")를 처리하지 못해 발생했던 오류를, 명령어 토큰 재조합 로

직을 구현하여 완벽하게 해결했다. 이 수정으로 프로그램의 견고성과 명령 처리 능력이 향상되었다.

ii. 시스템 통합

BST의 효율적인 검색(O($\log N$))과 연결 리스트의 유연한 삽입/삭제 기능을 결합하여, 음악 데이터의 이중 관리와 플레이리스트의 동적 관리가 원활하게 이루어졌다.

B. 개선할 점

i. 메모리 관리 효율

*QPOP*이나 *MAKEPL* 명령 시, 원본 데이터 노드를 복사하여 새로운 노드를 만들고 이를 BST나 *PlayList*에 삽입하는 방식으로 구현했다. 이는 데이터 복제가 발생하며 메모리 사용량이 증가한다.

ii. 대안

모든 자료 구조가 단일 음악 객체의 포인터를 공유하는 방식으로 설계하고, *Manager*가 메모리 해제를 전적으로 담당하도록 구현했다면, 메모리 관리의 안전성과 효율성을 높일 수 있었을 것이다. 현재 방식은 메모리 해제 (delete) 시점에 대한 복잡성을 증가시킨다.

결론적으로, 프로젝트의 모든 필수 기능과 요구사항은 성공적으로 구현되었으며, 자료 구조의 실질적인 응용 능력을 입증할 수 있었다.