데이터구조설계

DS_Project2 보고서

이름: 양서은

학번: 2022202061

학과: 컴퓨터정보공학부

교수님: 최상호 교수님

1. Introduction

B+-Tree, Selection Tree, Heap을 사용하여 도서 대출 관련 업무를 관리하는 프로그램을 구현한다. 도서 정보에는 제목, 도서 분류 코드, 발행 연도, 대출 권수가 있다. 현재 존재하는 도서들 중 대출이 가능한 도서를 B+-Tree로 관리하며, 도서 대출이 불가능해진 도서들은 B+-Tree에서 삭제되고 Selection Tree에 저장된다. Selection Tree의 각 run은 Min Heap으로 구성되어 있고 도서 분류 코드에 따라 서로 다른 독립적인 Heap을 구성한다. 각 도서는 도서 분류 코드는 최대 대출 권수를 가지고 있다.

000	3
100	3
200	3
300	4
400	4
500	2
600	2
700	2

위의 표에 따라 최대 대출 권수가 정해져 있으며 임의의 도서가 해당되는 도서 분류 코드의 최대 대출 권수에 도달하면 B+-Tree에서 삭제되고 Selection Tree에 삽입이 된다. 예를 들어 도서 분류 코드가 100인 도서 a는 3권이 대출된다면 더 이상 대출할 수 없고 Selection Tree로 도서 정보가 이동한다.

1) B+-Tree

B+-Tree는 자기 균형 이진 트리 데이터 구조로 정렬된 데이터를 유지하고 삽입, 삭제, 검색을 빠르게 수행할 수 있다. 이번 프로젝트의 경우 차수를 3으로 고정했기 때문에 각 index node는 최대 2개의 key 값과 3개의 자식 노드를 가리킬 수 있다.

B+-Tree는 index node, data node 2가지 종류의 노드를 가지고 있다. index node는 B+-Tree의 internal node로 key 값만 저장하고 있다. 따라서 삽입 및 삭제할 때 key 값을 기준으로 자식 노드를 판별한다. data node는 B+-Tree의 external node로 key 값과 element 값을 가지고 있다. 프로젝트에서 생성된 B+-Tree의 경우 key 값은 도서명, element 값은 해당 도서의 정보들이다. 따라서 datanode는 도서 정보들이 저장되어 있는 노드의 포인터가 element 값이 된다.

B+-Tree는 노드에 key값이 최대 개수를 초과하면 스스로 노드를 split하면서 재정렬을 하는데 따라서 모든 datanode들은 같은 레벨에 위치할 수 있다. 따라서 datanode들은 서로 양방향 연결될 수 있고 그렇게 함으로써 데이터 탐색 및 출력이 용이해진다.

2) Selection Tree

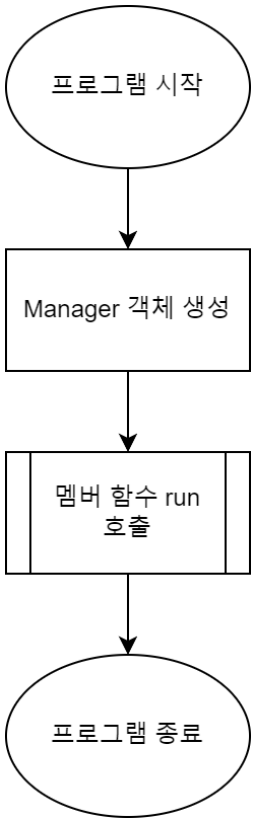
선택 트리로 Tournament Tree로 불리기도 한다. 트리의 최하단 노드들은 각 run들과 연결되어 있고 각 run에서 올라온 정보들을 토대로 임의의 기준에 따라 sibling node 중 1개를 취사선택해 부모 노드에 저장하며 root까지 올라간다. 이번 프로젝트의 경우 도서명을 기준으로 알파벳 순으로 Min Winner Tree를 구성한다. 따라서 알파벳 순으로 제일 작은 도서가 winner가 된다.

3) Heap

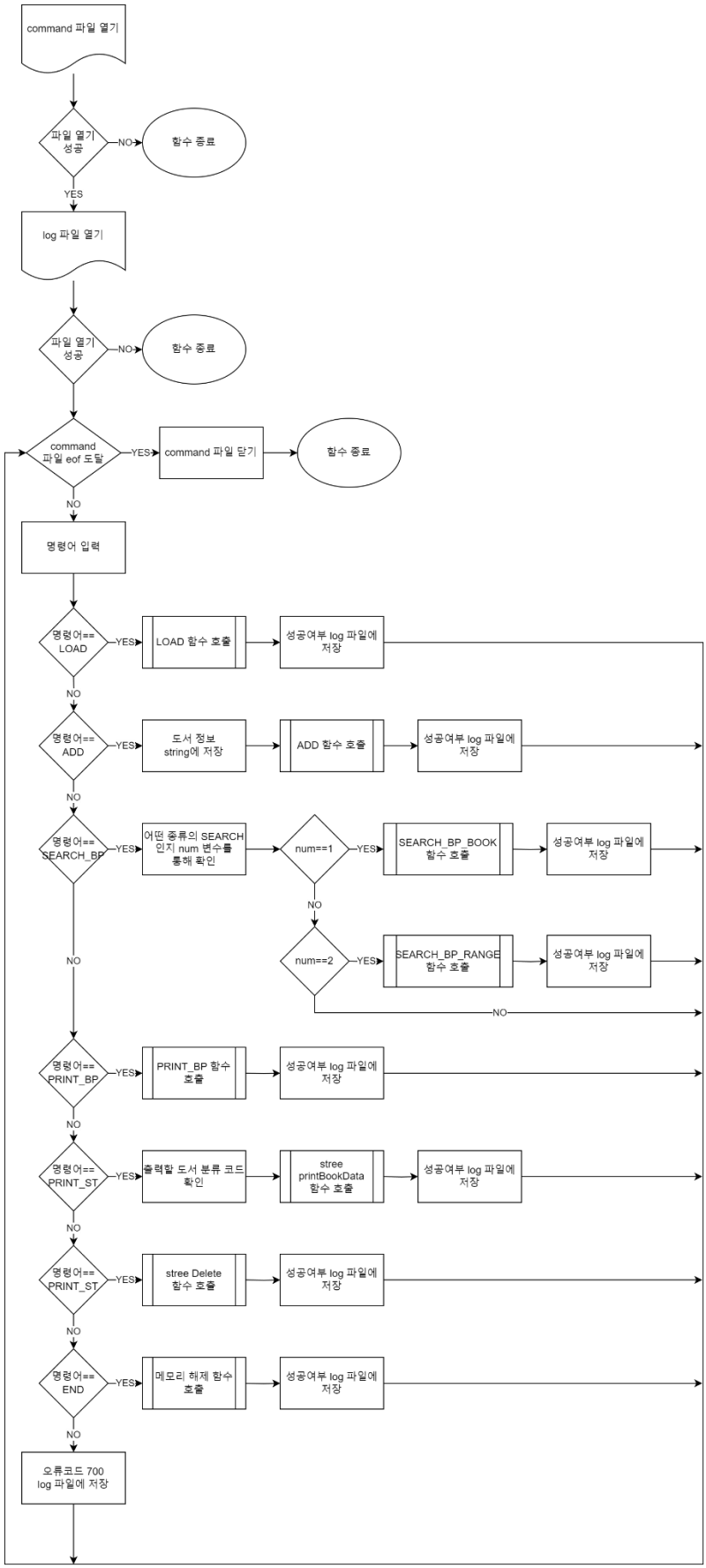
Binary Heap은 이진 트리로 부모 노드와 자식 노드가 특정 순서를 만족하는 트리이다. 이번 프로젝트에선 Min Heap으로 구현했기 때문에 모든 부모 노드는 항상 자식 노드보다 key 값이 작은 형태를 만족한다. 완전 이진 트리를 만족하고 노드가 추가, 삭제될 때마다 재정렬한다. 노드 추가의 경우 완전 이진 트리를 만족하게 최하단 노드에 추가 후 heapifyUp을 통해 재정렬하고, 삭제의 경우 root를 삭제하고 최하단 노드를 root로 올린 후 heapifyDown을 통해 재정렬한다. Heap의 경우 BST와 다르게 부모 자식 간의 크기 비교만 하기 때문에 sibling 노드의 크기 비교는 사용자가 정의해주지 않는 이상 대소 비교를 할 수 없다.

2. Flowchart

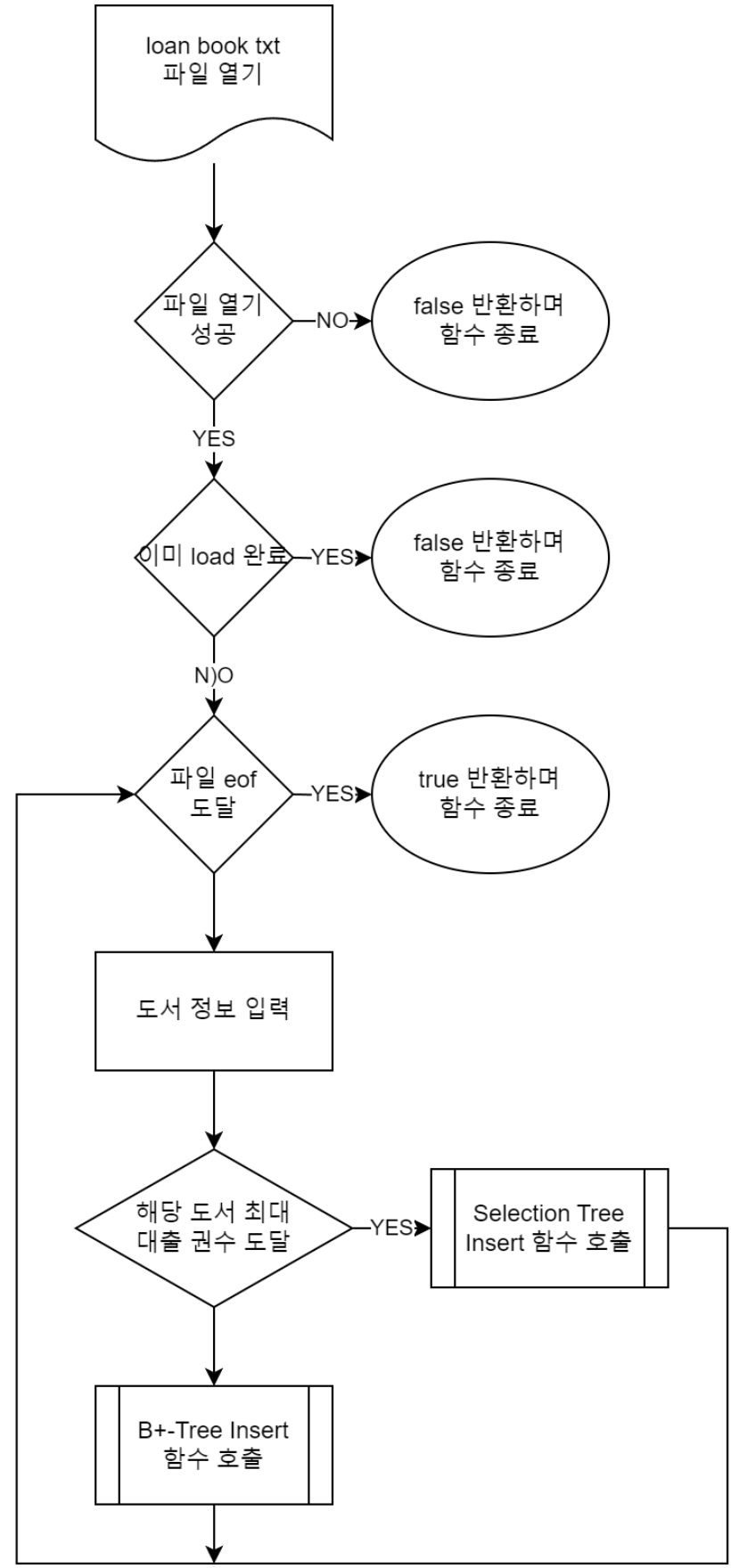
main 함수



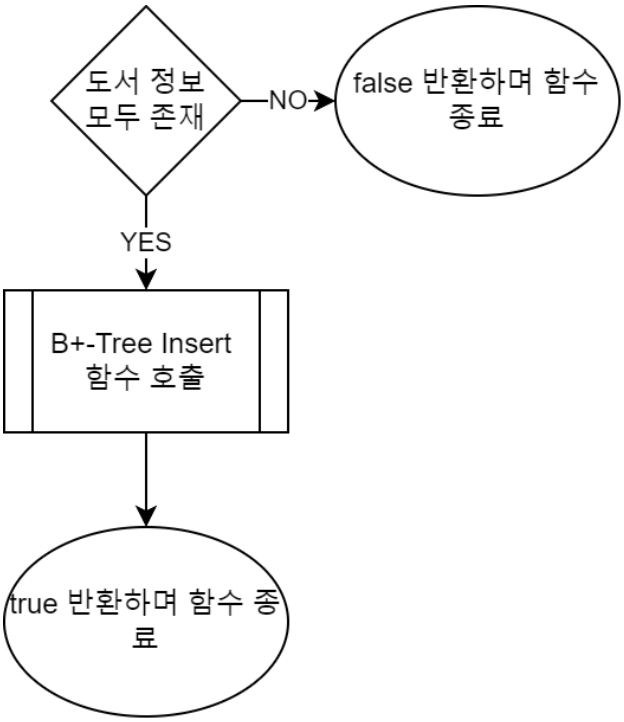
run 함수


command 파일 열기
파일 열기 성공
NO
함수 종료
YES
log 파일 열기
파일 열기 성공
NO
함수 종료
command 파일 eof 도달
YES
command 파일 닫기
함수 종료
NO
명령어 입력
명령어== LOAD
YES
LOAD 함수 호출
성공여부 log 파일에 저장
NO
명령어== ADD
YES
도서 정보 string에 저장
ADD 함수 호출
성공여부 log 파일에 저장
NO
명령어== SEARCH_BP
YES
어떤 종류의 SEARCH인지 num 변수를 통해 확인
num==1
YES
SEARCH_BP_BOOK 함수 호출
성공여부 log 파일에 저장
NO
num==2
YES
SEARCH_BP_RANGE 함수 호출
성공여부 log 파일에 저장
NO
NO
명령어== PRINT_BP
YES
PRINT_BP 함수 호출
성공여부 log 파일에 저장
NO
명령어== PRINT_ST
YES
출력할 도서 분류 코드 확인
stree printBookData 함수 호출
성공여부 log 파일에 저장
NO
명령어== PRINT_ST
YES
stree Delete 함수 호출
성공여부 log 파일에 저장
NO
명령어== END
YES
메모리 해제 함수 호출
성공여부 log 파일에 저장
NO
오류코드 700 log 파일에 저장

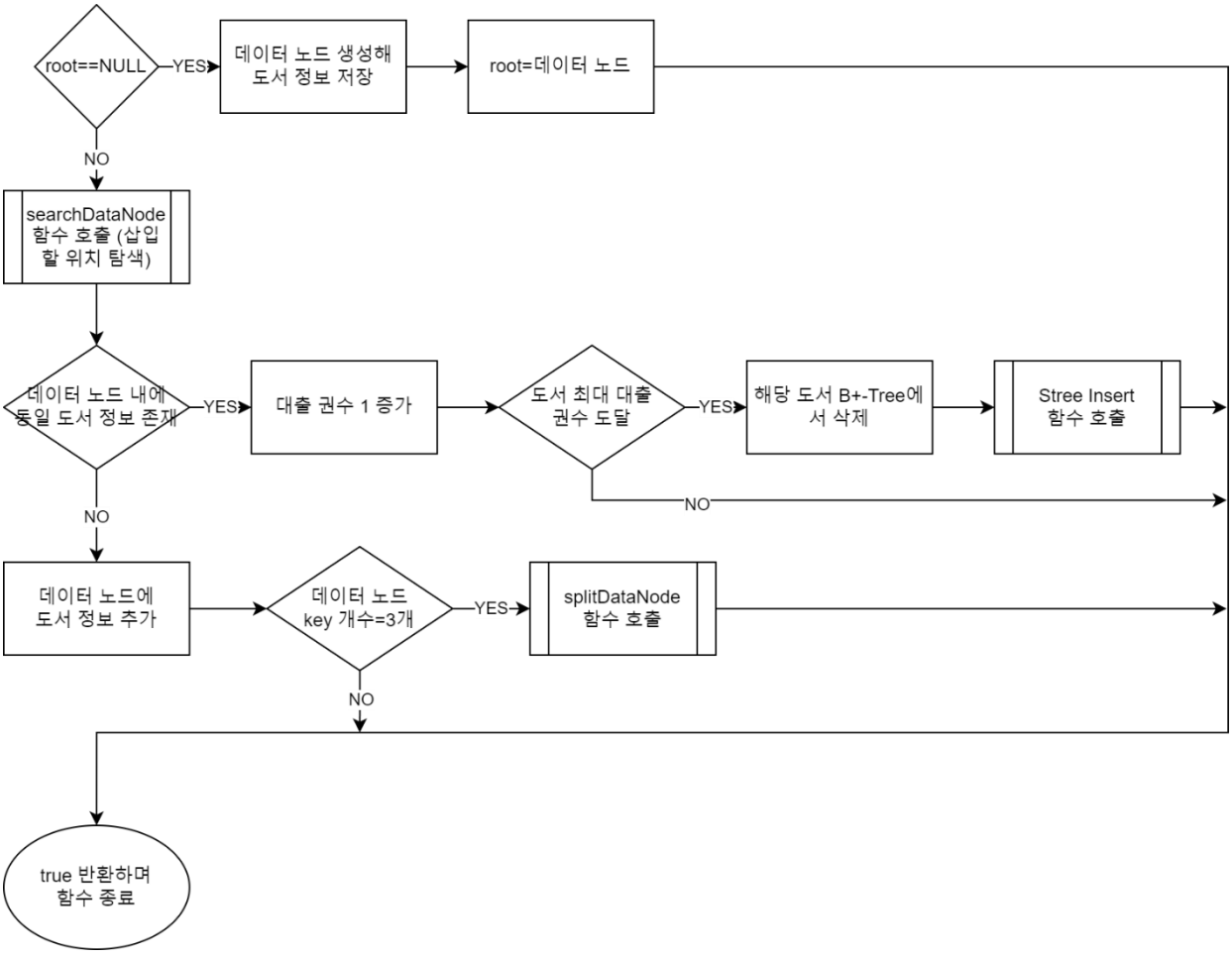
LOAD 함수


loan book txt
파일 열기
파일 열기
성공
NO
false 반환하며
함수 종료
YES
이미 load 완료
YES
false 반환하며
함수 종료
N)O
파일 eof
도달
YES
true 반환하며
함수 종료
도서 정보 입력
해당 도서 최대
대출 권수 도달
YES
Selection Tree
Insert 함수 호출
B+-Tree Insert
함수 호출

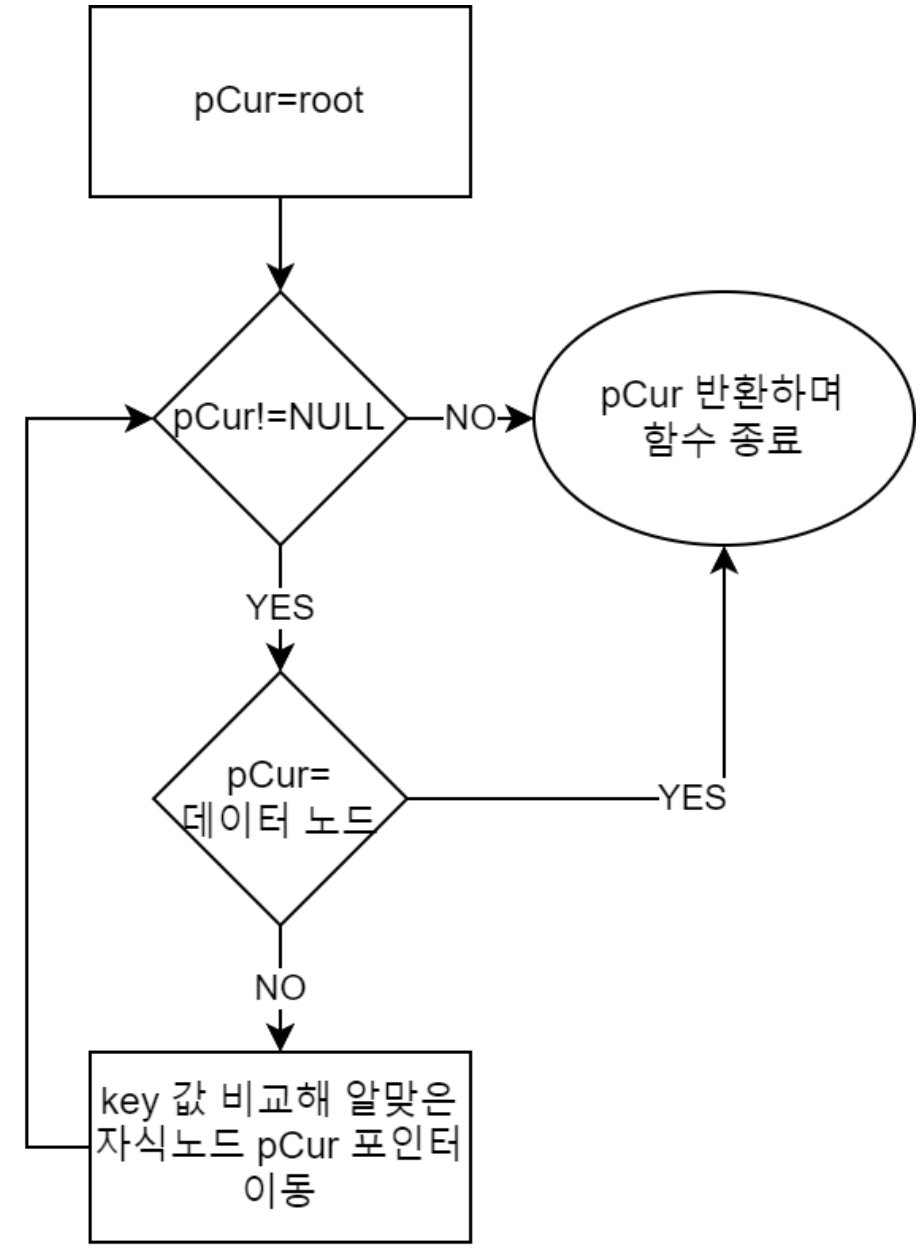
ADD 함수



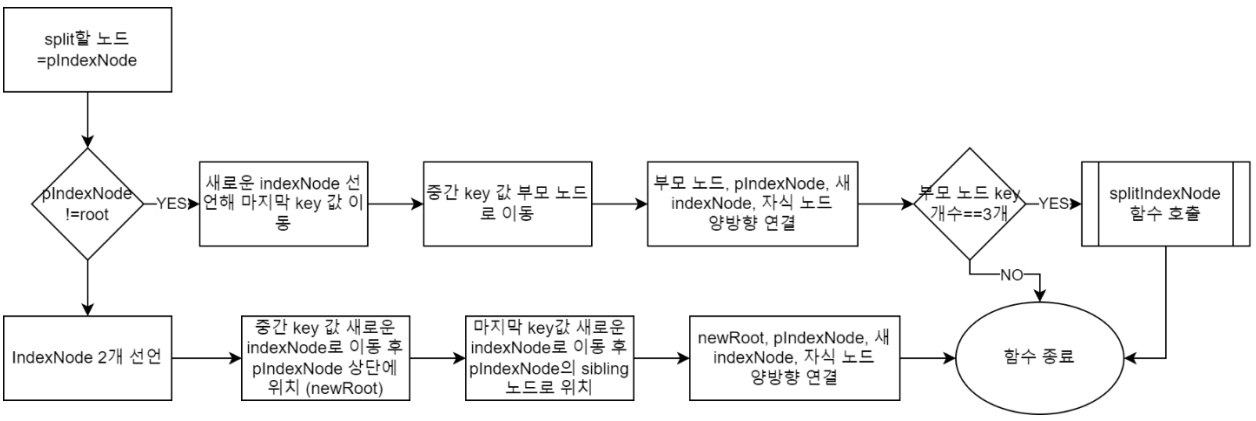
BpTree Insert 함수



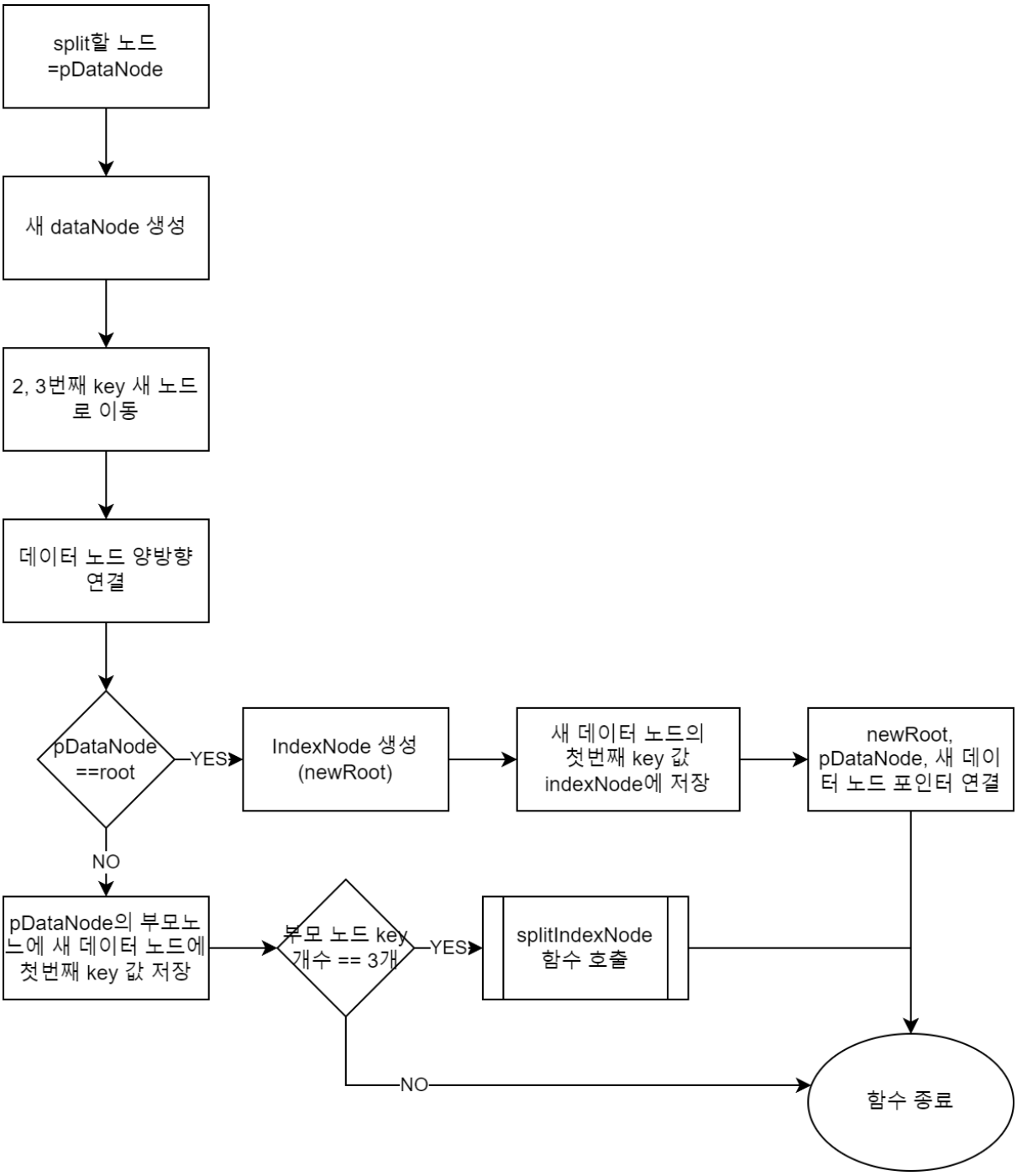
BpTree searchDataNode 함수



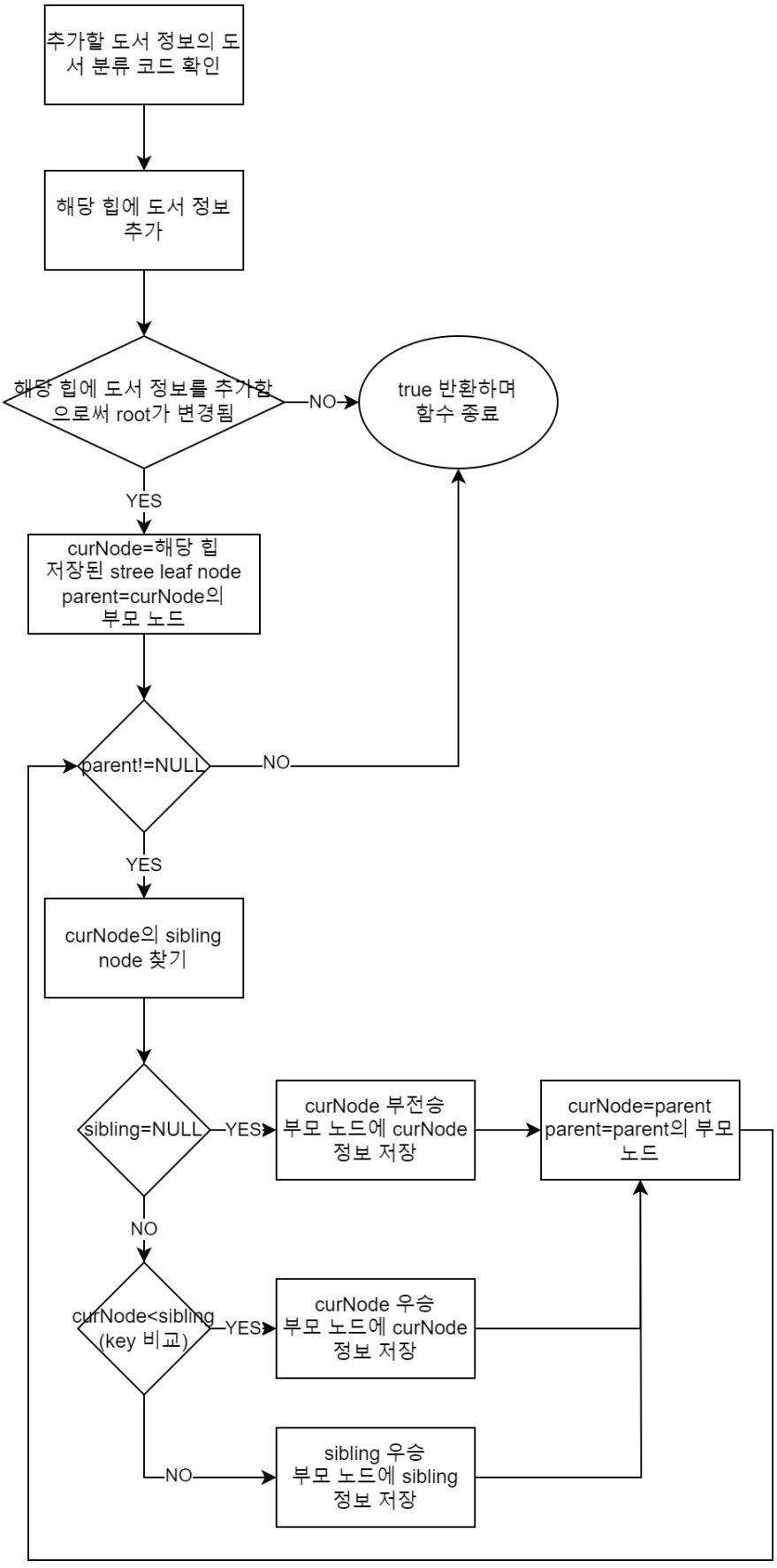
BpTree splitIndexNode 함수



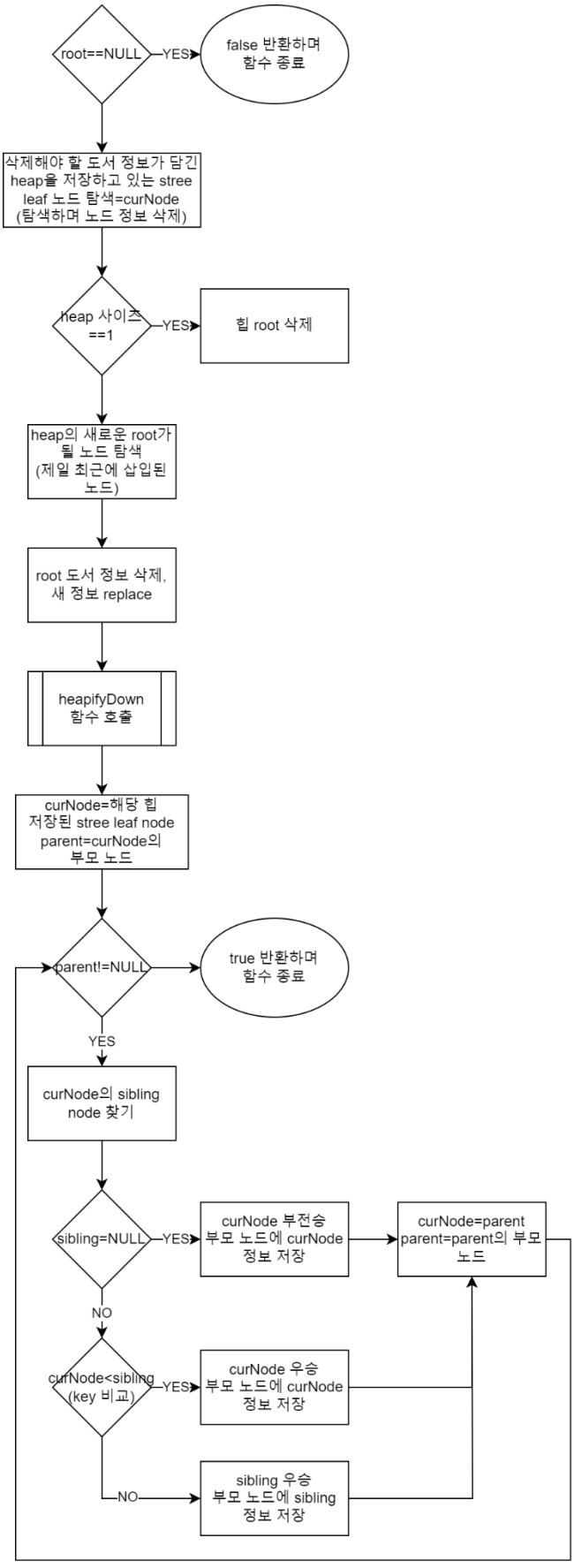
BpTree splitDataNode 함수



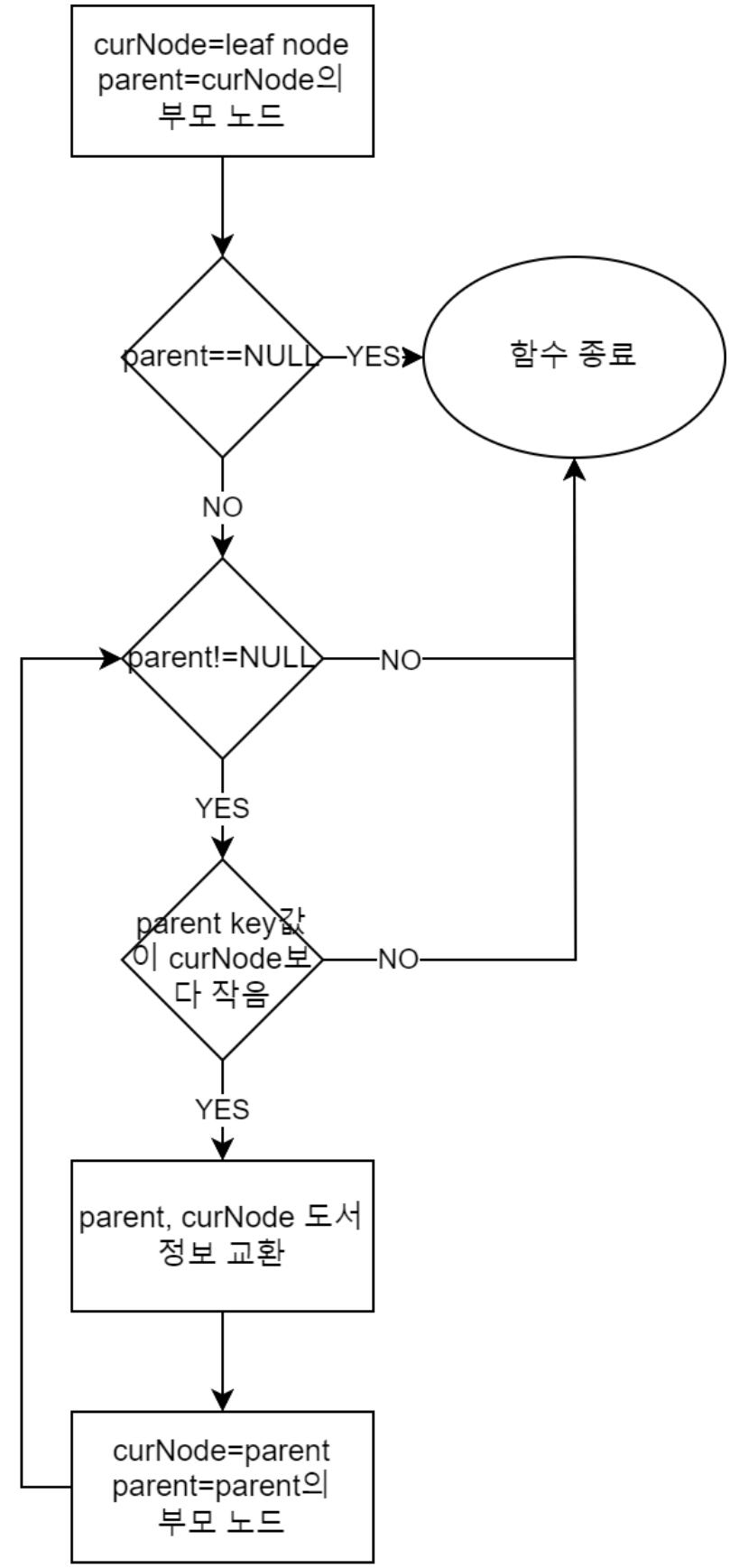
SelectionTree Insert 함수



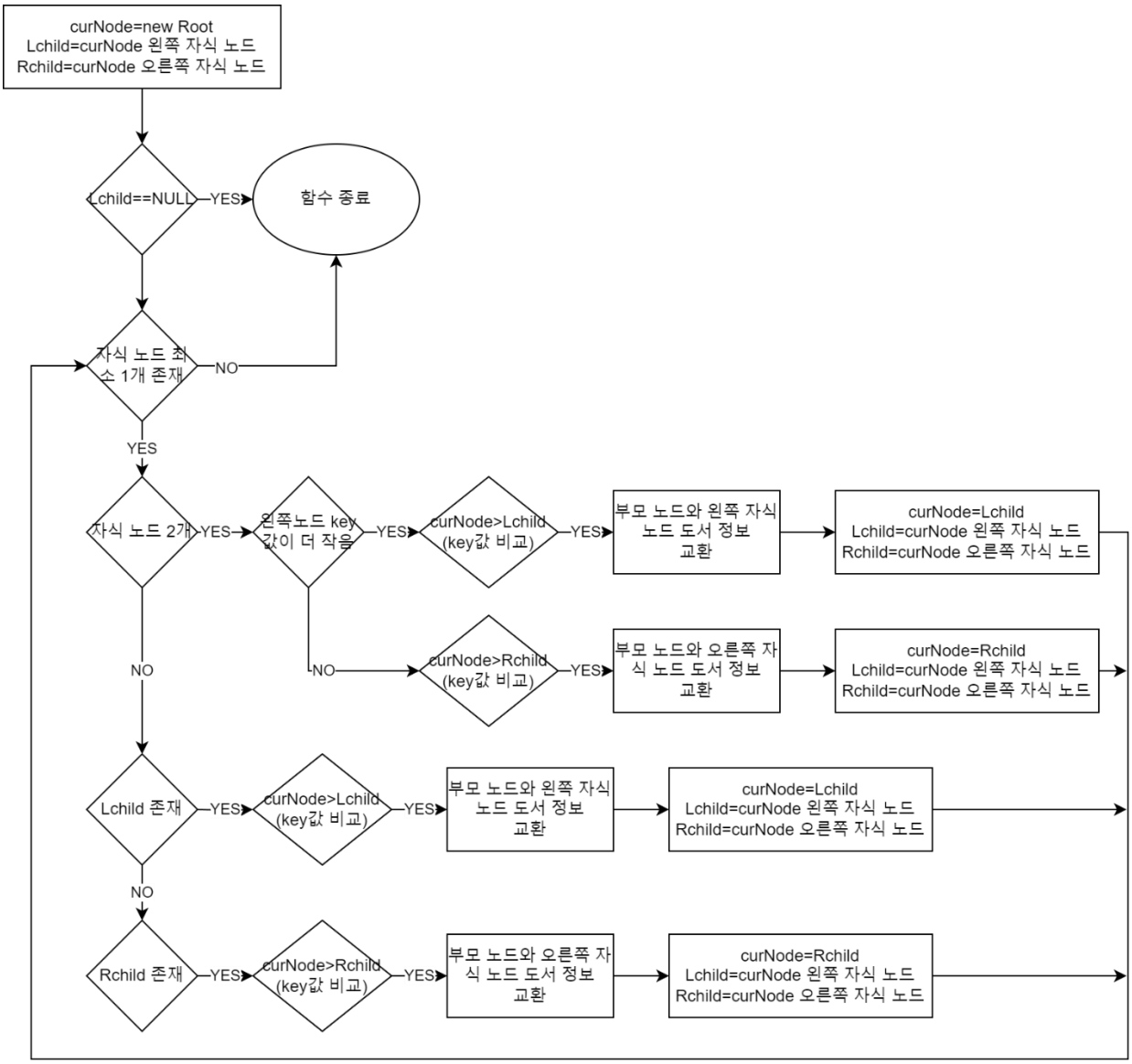
SelectionTree Delete 함수



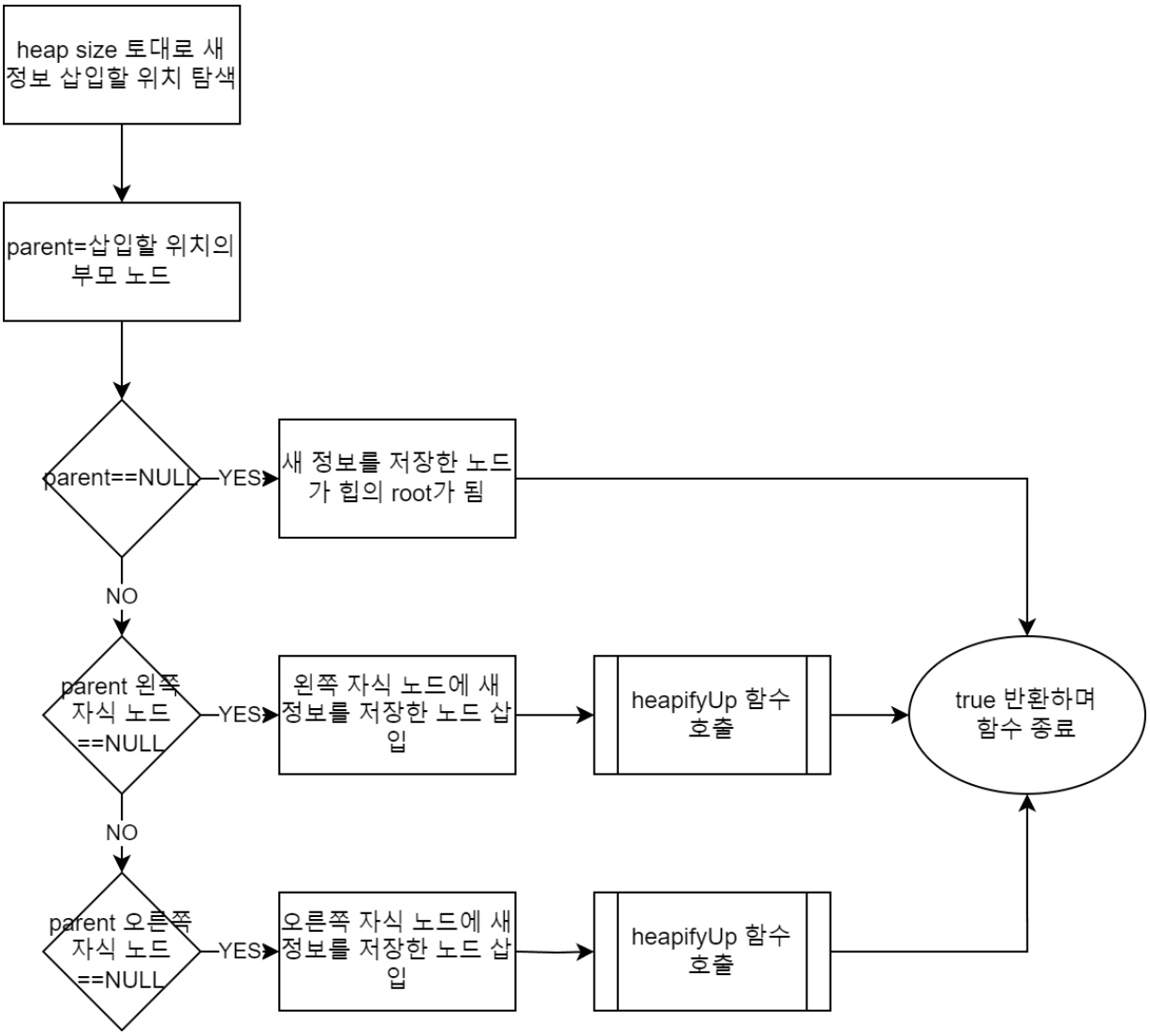
LoanBookHeap heapifyUp 함수



LoanBookHeap heapifyDown 함수



LoanBookHeap Insert 함수



3. Algorithm

- B+-Tree

  1) Insert

  첫 삽입의 경우 데이터 노드를 생성하여 도서 정보를 저장한다.
  첫 삽입이 아닐 경우 도서 정보를 저장할 데이터 노드를 찾는다. bptree의 root부터 도서명의 크기를 key값과 비교해 알맞은 포인터로 이동하며 찾는다. 만약 이미 bptree에 존재하던 책이라면 대출 권수를 증가시킨다. 증가시킨 후 해당 도서가 최대 대출 권수에 도달했다면 bptree에서 삭제 후 selection tree로 도서 정보를 이동한다. 새로운 도서라면 찾았던 데이터 노드에 도서 정보를 삽입한다. 만약 해당 데이터 노드의 key값이 최대 개수에 도달했다면 data node split을 진행한다. 새로운 data node를 생성한 후 첫번째 key 값을 제외한 key와 element를 새로운 data node로 옮겨준다.
  데이터 노드의 부모 노드가 존재하지 않을 경우, index node를 선언해준 후 새로 생성했던 data node의 첫번째 key 값을 저장한다. 그 후 모든 노드를 양방향 연결해주며 insert를 완료한다.
  데이터 노드의 부모 노드가 존재한다면 새로 생성했던 data node의 첫번째 key 값을 부모 노드에 삽입한다. 모든 노드를 양방향 연결해준 후 부모 노드의 key 값 개수를 확인해 split이 필요하다면 index node split을 진행한다.
  split할 index node가 root라면 새로운 index node를 2개 생성하여 하나는 새로운 root가 되고 split한 index node 2개는 새로운 root의 자식 노드가 된다. split할 노드가 가지고 있던 3개의 key 값 중 두번째 key값을 root 노드로 이동하고 마지막 key 값을 sibling 노드로 이동한다. 각 노드를 서로 알맞게 연결해주며 split을 종료한다.
  split할 index node가 root가 아니라면 새로운 index node를 생성해 마지막 key 값을 이동시킨다. 중간 key 값은 split을 진행하는 노드의 부모 노드로 이동한다. 새 index node 기준 부모, 자식 노드들과 모두 양방향으로 연결해주고 부모 노드의 key 값 개수를 확인한다. split이 필요하다면 재귀적으로 B+-Tree 구조를 만족할 때까지 index node split을 진행한다.

- Selection Tree

  1) Insert

  Selection Tree는 일반적인 트리와는 수행하는 기능이 다르다. 일반적인 트리는 일련의 정보를 임의의 기준에 따라 분류해 저장한다면 Selection Tree는 임의의 정보들 중 기준에 맞는 최적의 정보를 내놓는다. 이 프로그램에서 Selection Tree의

insert는 도서 정보를 받아와 해당 도서 분류 코드에 맞는 run을 찾아 저장하고 그에 따라 다시 Selection Tree를 재정렬한다.
bptree에서 삭제된 도서 정보가 stree로 넘어온다. 해당 도서의 분류 코드를 확인한 후 run인 heap에 저장한다. 이 때 heap의 root 정보가 바뀌지 않는다면 Selection tree를 재정렬할 필요가 없어진다.
재정렬이 필요하다면 새로운 노드가 저장된 heap의 정보를 가지고 있는 Selection tree의 leaf 노드에서부터 정렬이 시작된다. sibling node와 비교해 key값이 작은 노드의 정보가 부모 노드에 저장된다. root까지 비교를 진행하며 정렬을 완료한다.

2) Delete

이 프로그램에서 Selection Tree의 Delete는 현재 트리에 저장되어 있는 winner 정보를 삭제하고 다시 정렬하는 기능이다. root에서부터 도서 정보를 삭제해가며 해당 도서 정보가 저장되어 있던 heap까지 도달한다. heap의 root를 삭제한 후 heapifyDown을 진행해 heap 재정렬을 완료한 뒤 Selection Tree 재정렬을 진행한다. 자신(노드)과 sibling node와 비교해 NULL이면 자신의 정보가 부모 노드에 저장되고, NULL이 아니라면 key 값 대소 비교를 통해 알맞은 값을 부모 노드에 저장한다. root까지 도달했을 때 함수가 종료된다.

- Heap

1) Insert

힙의 경우 complete binary tree 형태를 만족하기 때문에 새로운 노드를 삽입할 위치가 트리의 사이즈에 따라 정해져 있다. 트리의 사이즈를 이진수로 변환 후 오른쪽에서 2번째 자리수부터 root에서 시작해 0이면 왼쪽 자식 노드, 1이면 오른쪽 자식 노드로 이동하며 삽입할 위치를 탐색한다. 해당 위치에 새로운 도서 정보 노드를 삽입한 후 heapifyUp 과정을 통해 힙을 재정렬한다.

2) HeapifyUp

힙에 새로운 노드가 저장되었을 때 Min heap 관계를 만족할 수 있도록 재정렬하는 기능이다. 새로 추가된 노드에서부터 root를 향해 나아가는데 해당 노드와 부모 노드의 대소 비교를 통해 자식 노드의 key 값이 더 작다면 부모 노드와 서로 자리를 바꾼다. 반복문을 통해 root까지 비교가 완료되었거나 비교 도중 이미 Min heap 관계를 만족한다면 빠져나오며 함수가 종료된다.

3) HeapifyDown

새로운 노드를 삽입했을 때 호출되는 heapifyUp과 반대로 힙에서 노드가 삭제되

었을 때 호출되는 기능이다. 힙의 경우 노드를 삭제할 때 항상 root 노드를 삭제한다. 그 후 트리에서 가장 마지막에 삽입되었던 노드를 root에 위치시킨 후 자식 노드와 비교하며 힙을 재정렬한다. 자식 노드가 2개이고 모두 부모 노드보다 작을 땐 두 자식 노드 중 더 작은 값과 도서 정보를 교환한다. 만약 자식 노드와 한 개라면 그 노드와 대소 비교를 통해 교환을 진행한다. 더 이상 비교할 자식 노드가 없거나 Min heap을 만족할 때 함수가 종료된다.

4. Result Screen

command txt 파일

```
LOAD
ADD quantum Dreams 500 Jackson Foster 2020
ADD quantum Dreams 500 Jackson Foster 2020
ADD quantum Dreams 500 Jackson Foster
ADD great Expectations 000 Charlotte Bronte 2002
ADD fahrenheit 451 100 Charles Dickens 1980
ADD slaughterhouse-Five 600 Mary Shelley 1968
ADD the Odyssey 700 Ray Bradbury 1913
PRINT_BP
SEARCH_BP quantum Dreams
SEARCH_BP jane Eyre
SEARCH_BP a
SEARCH_BP gardens Holding Time
SEARCH_BP a e
SEARCH_BP m p
PRINT_ST 300
PRINT_ST 400
PRINT_ST 500
PRINT_ST 700
DELETE
DELETE
PRINT_ST 300
PRINT_ST 400
PRINT_ST 500
PRINT_ST 700
DELETE
DELETE
DELETE
DELETE
DELETE
EXIT
```

loan_book txt 파일

the Enchantment of Language	000	Emma Richardson	2018	0	
gardens Holding Time	100	Alex Kennedy	2021	0	
fundamentals of Machine Learning	200	David Miller	2019	0	
philosopher's Morning	100	Sophia Anderson	2020	0	
echoes of Eternity	300	Benjamin Harper	2022	0	
harmony in Chaos	400	Olivia Williams	2017	0	
quantum Dreams	500	Jackson Foster	2020	0	
canvas of the Cosmos	600	Maya Turner	2019	0	
infinite Horizons	100	Liam Parker	2021	0	
wings of Imagination	700	Harper Collins	2018	0	
sculpting the Mind	300	Victoria Lane	2016	4	
journey to the Unknown	500	Ethan Brooks	2020	0	
rhythms of Nature	300	Isabella Hayes	2015	4	
beyond the Veil	300	Gabriel Knight	2023	4	
the Quantum Code	400	Mia Sullivan	2018	0	
celestial Harmony	600	Adrian Lee	2022	0	
crime and Punishment	500	Louisa May Alcott	2002	0	
madame Bovary	500	Gustave Flaubert	1960	0	
the Lord of the Rings	000	J.R.R. Tolkien	1908	0	
brave New World	500	Charlotte Bronte	1926	2	
pride and Prejudice	500	Jane Austen	1979	0	
jane Eyre 700	Anthony Burgess	2014	0		
the Grapes of Wrath	300	Charles Dickens	1958	0	
a Clockwork Orange	000	Joseph Conrad	1939	0	
the Adventures of Huckleberry Finn	700	Kurt Vonnegut	1939	0	
a Clockwork Orange	500	Charles Dickens	1986	2	
great Expectations	300	Homer	1997	0	
the Iliad 400	Aldous Huxley	1990	0		
the Lord of the Rings	200	Gustave Flaubert	1959	0	
to Kill a Mockingbird	200	Gustave Flaubert	1964	0	
great Expectations	300	Mary Shelley	1987	0	
fahrenheit 451	500	Fyodor Dostoevsky	1925	0	
great Expectations	000	Charlotte Bronte	2002	0	
wuthering Heights	300	Gustave Flaubert	1916	0	

command	log txt file
LOAD	========LOAD======== Success ========== 성공적으로 loan_book.txt 파일을 열어 도서 정보를 manager 객체에 저장함
ADD quantum Dreams 500 Jackson Foster 2020 ADD quantum Dreams 500 Jackson Foster 2020 ADD quantum Dreams 500 Jackson Foster ADD great Expectations 000 Charlotte Bronte 2002 ADD fahrenheit 451 100 Charles Dickens 1980 ADD slaughterhouse-Five 600 Mary Shelley 1968 ADD the Odyssey 700 Ray Bradbury 1913	========ADD======== quantum Dreams/500/Jackson Foster/2020 ========== ========ADD======== quantum Dreams/500/Jackson Foster/2020 ========== ========ERROR======== 200 ========== ========ADD======== great Expectations/000/Charlotte Bronte/2002 ========== ========ADD======== fahrenheit 451/100/Charles Dickens/1980 ========== ========ADD======== slaughterhouse-Five/600/Mary Shelley/1968 ========== ========ADD======== the Odyssey/700/Ray Bradbury/1913 ========== 도서 정보를 토대로 manager 객체에 저장함 3번째 ADD의 경우 도서 정보 중 발행 연도가 누락되어 ADD가 진행되지 않음

PRINT_BP	======PRINT_BP====== a Clockwork Orange/000/Joseph Conrad/1939/0 canvas of the Cosmos/600/Maya Turner/2019/0 celestial Harmony/600/Adrian Lee/2022/0 crime and Punishment/500/Louisa May Alcott/2002/0 echoes of Eternity/300/Benjamin Harper/2022/0 fahrenheit 451/500/Fyodor Dostoevsky/1925/1 fundamentals of Machine Learning/200/David Miller/2019/0 gardens Holding Time/100/Alex Kennedy/2021/0 great Expectations/300/Homer/1997/3 harmony in Chaos/400/Olivia Williams/2017/0 infinite Horizons/100/Liam Parker/2021/0 jane Eyre/700/Anthony Burgess/2014/0 journey to the Unknown/500/Ethan Brooks/2020/0 madame Bovary/500/Gustave Flaubert/1960/0 philosopher's Morning/100/Sophia Anderson/2020/0 pride and Prejudice/500/Jane Austen/1979/0 slaughterhouse-Five/600/Mary Shelley/1968/0 the Adventures of Huckleberry Finn/700/Kurt Vonnegut/1939/0 the Enchantment of Language/000/Emma Richardson/2018/0 the Grapes of Wrath/300/Charles Dickens/1958/0 the Iliad/400/Aldous Huxley/1990/0 the Lord of the Rings/000/J.R.R. Tolkien/1908/1 the Odyssey/700/Ray Bradbury/1913/0 the Quantum Code/400/Mia Sullivan/2018/0 to Kill a Mockingbird/200/Gustave Flaubert/1964/0 wings of Imagination/700/Harper Collins/2018/0 wuthering Heights/300/Gustave Flaubert/1916/0 ==========
SEARCH_BP quantum Dreams SEARCH_BP jane Eyre	========ERROR======== 300 ========== ========== ========SEARCH_BP======== jane Eyre/700/Anthony Burgess/2014/0 ========== ========== quantum Dreams 도서의 경우 최대 대출 권수에 도달하여 stree로 이동했기 때문에 bptree에선 찾을 수 없음
SEARCH_BP a SEARCH_BP gardens Holding Time SEARCH_BP a e SEARCH_BP m p	========ERROR======== 300 ========== ========== ========SEARCH_BP======== gardens Holding Time/100/Alex Kennedy/2021/0 ========== ========== ========SEARCH_BP======== a Clockwork Orange/000/Joseph Conrad/1939/0 canvas of the Cosmos/600/Maya Turner/2019/0 celestial Harmony/600/Adrian Lee/2022/0 crime and Punishment/500/Louisa May Alcott/2002/0 echoes of Eternity/300/Benjamin Harper/2022/0 ========== ========== ========SEARCH_BP======== philosopher's Morning/100/Sophia Anderson/2020/0 pride and Prejudice/500/Jane Austen/1979/0 ========== ========== a라는 제목을 가진 도서를 찾을 수 없음

PRINT_ST 300 PRINT_ST 400 PRINT_ST 500 PRINT_ST 700	========PRINT_ST======== beyond the Veil/300/Gabriel Knight/2023/4 rhythms of Nature/300/Isabella Hayes/2015/4 sculpting the Mind/300/Victoria Lane/2016/4 ========== ========ERROR======== 500 ========== ========PRINT_ST======== a Clockwork Orange/500/Charles Dickens/1986/2 brave New World/500/Charlotte Bronte/1926/2 quantum Dreams/500/Jackson Foster/2020/2 ========== ========ERROR======== 500 ========== stree에 저장되어 있는 도서를 출력 400, 700에 해당하는 도서들은 모두 bptree에 저장되어 있음
DELETE DELETE	========DELETE======== Success ========== ========DELETE======== Success ==========
PRINT_ST 300 PRINT_ST 400 PRINT_ST 500 PRINT_ST 700	========PRINT_ST======== rhythms of Nature/300/Isabella Hayes/2015/4 sculpting the Mind/300/Victoria Lane/2016/4 ========== ========ERROR======== 500 ========== ========PRINT_ST======== brave New World/500/Charlotte Bronte/1926/2 quantum Dreams/500/Jackson Foster/2020/2 ========== ========ERROR======== 500 ========== 2번 DELETE를 진행한 후 다시 stree print를 진행 500번대 도서 1개, 300번대 도서 1개 삭제됨

DELETE DELETE DELETE DELETE DELETE	========DELETE======== Success ==================== ========DELETE======== Success ==================== ========DELETE======== Success ==================== ========DELETE======== Success ==================== ========ERROR======== 600 ==================== stree DELETE 진행 모든 도서 정보가 삭제된 후에도 DELETE 명령이 들어올 경우 ERROR 코드 출력
EXIT	 ========EXIT======== Success ==================== 프로그램 종료

5. Consideration

B+-Tree의 경우 수업 시간에 배울 때도 이해가 원활하게 되지 않던 데이터구조라 C++로 짜려고 하니 너무 막막했다. 하지만 스켈레톤 코드를 통해 class의 멤버 변수를 이해하고 멤버 함수를 적극적으로 활용하여 코딩하니 구현할 수 있었다. split 구현이 가장 까다로웠는데 노드를 2개로 분할하면서 부모, 자식 노드와 양방향 연결을 새로 해주는 과정에서 예외가 많이 발생했다. 하지만 각 경우의 수를 하나하나 고민하며 노드 연결을 해주며 수정할 수 있었다. 이 과정에서 디버깅의 중요성을 깨달았는데 어떻게 코드를 짜는가에 따라 root 노드에서부터 연결은 잘 되어 있어도 data node로부터 root까지 올라갈 때 역방향 연결이 제대로 되어 있지 않을 수 있음을 깨달았다.

B+-Tree를 구현하면서 이번 프로젝트는 차수를 3으로 고정했지만 class 자체는 map의 원소를 포인터로 선언함으로써 차수가 다른 B+-Tree도 만들 수 있는 class이다. 구현하면서 항상 map의 size가 3임을 가정하고 멤버 함수를 작성했는데 멤버 함수들도 차수에 따라 operation이 바뀔 수 있게 수정한다면 충분히 차수가 바뀌더라도 정상적으로 동작하는 프로그램을 만들 수 있을 것이다. 기회가 된다면 꼭 구현해보고 싶다.

모든 기능을 구현하고 메모리 누수를 방지하기 위해 프로그램을 종료하면서 모든 노드를 할당 해제해주었는데 그럼에도 불구하고 누수가 발생했다. 모든 할당 해제 함수를 체크하며 수정하였음에도 누수가 발생하여 고민하던 중 Selection Tree의 Delete 함수에서 heap을 재정렬 할 때 root의 도서 정보를 할당 해제하지 않고 바로 다른 도서 정보를 덮어씌운 것을 확인할 수 있었다. 따라서 현재 root에 저장되어 있는 도서 정보를 할당 해제하고, 새로운 도서 정보 노드와 연결시키도록 코드를 수정해 모든 메모리 누수를 방지할 수 있게 되었다.