

Contents

A. Introduction	3
1. 프로젝트 목표	3
2. 문제 정의	3
3. 문제 해결 방법 고찰	4
B. Body	5
1. 구현	5
개발 환경	5
자료구조	5
3 level hierarchy	10
세부적인 구현	11
Input	14
Output	15
2. 결과	16
예매 & 취소	16
잘못된 입력	19
C. Conclusion	20
가정	20
제한	20
결론	21

Figure Contents

Figure 1 RBT의 node	6
Figure 2 RBT.....	6
Figure 3 int ***movie가 담고 있는 내용	9
Figure 4 주요 함수와 자료구조	10
Figure 5 input(예매)	14
Figure 6 input(취소)	14
Figure 7 예매 output.....	15
Figure 8 취소 output.....	15
Figure 9 Day1 movie 1 reserve (day1-1 reserve.png).....	16
Figure 10 Day 1 movie 1 reservation check(day1-1 reserve check.png) ...	17
Figure 11 취소 후 확인(day1-1 cancel.png)	17
Figure 12 범위를 벗어난 입력.....	19
Figure 13 범위를 벗어난 입력2.....	19

A. Introduction

1. 프로젝트 목표

Red Black Tree 및 Algorithms 과목에서 배운 내용을 토대로 영화 예매 프로그램을 작성한다.

2. 문제 정의

1. 입력은 두 가지 경우로 나뉜다.
 - a) 예약: 영화 이름(혹은 번호), 상영 시간, 상영 날짜, 좌석 번호를 사용자로부터 입력 받아 예약을 진행한다.
 - b) 취소: 예약번호를 사용자로부터 입력 받아 예약을 진행한다.
2. 두 가지 입력에 대한 출력은 각각 다음과 같다.
 - a) 예약 번호, 영화 이름(혹은 번호), 상영 시간, 상영 날짜, 좌석 번호를 출력하며 예약이 완료되었음을 알린다.
 - b) 취소된 영화의 예약번호를 취소가 완료되었다는 문구와 함께 출력한다.
3. 영화의 종류는 3가지이다.
4. 하루에 3가지 영화 모두 3번 상영한다.
5. 최대 앞으로 3일까지 예약을 진행할 수 있다.
6. 각 영화마다 100개의 좌석이 있다.
7. 예약번호를 key로 하는 RBT를 사용하되 Node에 예약에 관련된 모든 정보를 저장해야 한다.
8. RB_INSERT, RB_DELETE 후에는 총 노드 수, root와 leaves(왼쪽부터)의 key 값, 트리의 높이를 출력해야 한다.
9. 프로그램은 총 3단계로 구성된다 – UI, submodule, primitive functions
10. 각 영화의 좌석 중 30%는 임의로 채워진 상태에서 예약을 받는다.
11. 27개의 영화를 적어도 1번씩은 test를 위해 예약하고 취소해야 한다.
12. 적절한 코멘트를 작성한다.
13. 역할에 따라 함수를 적절히 나눠 modularity를 높인다.

3. 문제 해결 방법 고찰

1. 사용자가 예약/취소/프로그램 종료 중 하나를 선택하도록 한다.
2. 사용자가 종료를 입력하기 전까지 계속 예약 혹은 취소를 진행한다.
3. 사용자가 예약을 선택하면 다음과 같이 수행한다.
 - a) 상영 날짜를 입력 받는다.
 - b) 그 날짜에 해당하는 영화 이름과 상영 시간이 적힌 3 x 3 표를 출력하고 1~9까지의 입력을 이용하여 사용자가 원하는 상영시간과 영화를 선택하도록 한다.
 - c) 선택된 영화의 좌석 현황을 10 x 10으로 나타낸다. 예약된 좌석은 X, 예약되지 않은 영화는 좌석 번호를 출력한다.
 - d) 만약 이미 예약된 좌석을 선택하면 이미 예약된 좌석이라는 문구와 함께 다시 좌석을 선택하도록 한다.
 - e) 모든 입력에 대해 범위를 벗어난 입력을 한다면 에러 문구를 출력하고 다시 입력을 받는다.
4. 사용자가 취소를 선택하면 다음과 같이 수행한다.
 - a) 예약 번호를 입력하도록 한다.
 - b) 만약 예약 번호에 해당하는 예약이 시스템 내에 없다면 에러 문구를 출력하고 다시 입력을 받는다.
 - c) 취소가 완료되면 취소한 예약 번호, 영화 이름, 상영 날짜, 상영 시간과 좌석번호를 출력한다.
5. 사용자가 요구한 사항을 완료하면 결과를 출력하고 다시 사용자로부터 입력을 받는다.
6. 예약 정보는 예약 번호를 key로 하는 node에 저장되며 모든 예약이 하나의 RBT에 저장된다.
7. 예약 후에는 해당 정보를 트리에 insert, 취소 후에는 해당 node를 트리에서 delete한다.
8. insert나 delete를 진행한 후에는 트리의 전체 노드 수, 높이, root의 키 값, leaves의 키 값을 출력한다.
9. random 예약을 위해 각 영화마다 30개의 좌석을 임의로 중복 없이 선택한 후 예약 정보를 tree에 저장한다.

B. Body

1. 구현

개발 환경

1. 사용언어: C
2. compiler: gcc 7.3.0 32 bit
3. IDE: code block 17.12
4. Encoding: UTF-8

자료구조

1. RBT

트리에에는 예약 번호를 key로 하는 노드를 저장한다. 모든 예약 정보는 하나의 트리에 저장한다. 예약 번호만을 입력하여 취소를 진행하는 과정에서 가장 쉽게 해당 노드를 찾을 수 있는 방식이기 때문이다.

만약 영화마다 혹은 상영날짜마다 다른 트리에 저장했다면 각각의 트리에서 그 예약 번호가 존재하는지 확인해야 할 것이다. 예를 들어 각 영화마다 트리를 따로 가진다고 생각해보자. 총 27개의 트리가 필요할 것이다. 각 트리마다 n개의 노드가 있다고 생각하면 트리마다 $\log(n)$ 만큼의 시간 복잡도를 갖는다. 최악의 경우 27개의 트리에 대해 search를 진행해야 하고 $27\log n$ 이 upper bound가 된다.

그러나 만약 모든 영화의 정보를 한 트리에 저장하고 각 영화마다 n개의 예약 정보를 담은 노드가 있다고 가정하면(앞서 영화마다 트리를 만든 경우와 총 node의 개수는 동일하다.) $\log(27n)$ 이 upper bound로 앞선 경우보다 매우 향상되었다고 할 수 있다.

RB_INSERT의 input argument에는 키 값 외에도 영화 번호, 좌석 정보 등도 포함된다. RB_DELETE는 트리 pointer를 제외하면 키 값만 입력하면 된다.

```

/**
 * struct for Red Black tree node
 */
typedef struct __rbtNode{
    int key;
    int movie;
    int time;
    int date;
    int seat;
    COLOR color;
    struct __rbtNode *left;
    struct __rbtNode *right;
    struct __rbtNode *p;
} RBT_NODE;

```

Figure 1 RBT의 node

트리의 노드에는 Figure 1에서 보이는 대로 키 값, 영화 번호, 상영 시간, 상영날짜, 좌석 번호와 RBT를 구성하기 위한 기본적인 값들이 저장된다.

```

/**
 * struct for Red Black Tree
 */
typedef struct __rbt{
    RBT_NODE *root;
    RBT_NODE *NIL;
    int num_node;
} RBT;

```

Figure 2 RBT

RBT를 사용자 지정 struct로 만들어 사용한다. 이 구조체에는 트리의 root의 주솟값, NIL의 주솟값이 포함되고 추가로 노드의 개수도 저장된다. 노드의 개수는 INSERT를 성공적으로 하였을 때 증가하고 DELETE를 성공적으로 하였을 때 감소한다.

RBT에 관련된 함수는 다음과 같다.

- a) RBT Tree_init(void)
트리의 nil node를 만들고 root와 연결시킨 후 tree의 pointer를 반환한다.
- b) void tree_free(RBT_NODE *node)

트리에 사용된 동적 할당 메모리를 free하는 역할을 한다. 재귀적으로 트리의 왼쪽과 오른쪽 child를 이 함수로 call하고 해당 subtree의 root를 free한다.

- c) void RB_INSERT(RBT *T, int key, int movie, int date, int time, int seat)
RBT의 insert 함수이다. key만을 추가로 하던 이전 과제에서의 구현과는 다르게 다른 예약 정보도 input으로 받으며 이를 이용하여 insert 함수 내에서 node를 만들어 트리에 추가한다. (key값이 현재 트리에 없는 경우에만) time에는 사용자가 입력한 영화 번호(1 ~ 9)를 저장한다. (노드로부터 영화 정보를 읽어 오기 위해서)
- d) void RB_INSERT_FIXUP(RBT *T, RBT_NODE *newnode)
RBT rule을 지키기 위해 사용되는 함수로 교재의 pseudo code를 참조하였다.
- e) void LEFT_ROTATE(RBT *T, RBT_NODE *x)
입력 받은 node를 왼쪽으로 회전한다.
- f) void RIGHT_ROTATE(RBT *T, RBT_NODE *x)
입력 받은 node를 오른쪽으로 회전한다.
- g) RBT_NODE *rb_search(RBT *T, int z)
트리 내에 z 값을 key로 갖는 node가 있다면 그 노드의 주소를 반환하고 없다면 NULL을 반환한다.
- h) void RB_TRANSPLANT(RBT *T, RBT_NODE *u, RBT_NODE *v)
입력 받은 두 노드의 자리를 변경한다.
- i) void RB_DELETE(RBT *T, int key)
입력 받은 값을 key로 가지는 노드가 있다면 이 노드를 삭제하고 그렇지 않다면 그런 노드는 존재하지 않는다는 문구를 출력한다.
- j) void RB_DELETE_FIXUP(RBT *T, RBT_NODE *x)
RBT rule을 지키기 위해 사용되는 함수로 교재의 pseudo code를 참조하였다.
- k) RBT_NODE *T_MIN(RBT *T, RBT_NODE *x)
입력 받은 node를 root로 취급하여 그 subtree의 최솟값을 반환한다.
- l) void Tree_print(RBT_NODE *node, RBT *T)
leaf인 노드(자손이 없는 노드)의 키 값과 색을 출력한다.
- m) char color(RBT_NODE *node)

enum으로 정의된 RED와 BLACK을 char 값으로 변환해 반환한다.

- n) `int Tree_height(RBT *T, RBT_NODE *cur)`
트리의 최대 높이를 출력한다. 재귀적으로 왼쪽 subtree와 오른쪽 subtree의 높이 중 높은 값에 1을 더해(본인의 높이) 반환한다.
- o) `int findMax(int a, int b)`
입력 받은 두 정수 중 큰 값을 반환한다. `Tree_height`에서 사용된다.

2. movie의 좌석 정보를 담은 array

movie의 정보를 담은 array이다. 총 27개 영화의 좌석 정보를 담고 있다. 영화의 좌석은 10 x 10 의 형태를 가지고 있으므로 각 영화당 2차원 배열을 갖는데 이런 영화가 27개가 있으므로 이 배열은 2차원 배열의 좌석 정보를 담은 array를 element로 하는 3차원 배열이다.

이 배열은 모두 동적할당을 받는다.

- a) `int ***movie_array(void)`
3차원 배열을 정의하고 0으로 초기화하여 이 배열의 첫번째 element의 주소를 반환한다.
- b) `void free_movie(int ***movie)`
3차원 배열에 사용된 동적 할당 메모리를 반환한다.

int ***movie	0	Day1	movie1	Time1
	1			Time2
	2			Time3
	3		Movie2	Time4
	4			Time5
	5			Time6
	6		Movie3	Time7
	7			Time8
	8			Time9
	9	Day2	Movie1	Time1
	10			Time2
	11			Time3
	12		Movie2	Time4
	13			Time5
	14			Time6
	15		Movie3	Time7
	16			Time8
	17			Time9
	18	Day3	Movie1	Time1
	19			Time2
	20			Time3
	21		Movie2	Time4
	22			Time5
	23			Time6
	24		movie3	Time7
	25			Time8
	26			Time9

Figure 3 int *movie가 담고 있는 내용**

movie는 Figure 3에서 표현된 대로 좌석의 정보를 저장하고 있다. (ex. movie[23]에는 day3 movie2 time 6-화면에 표시되는 기준, movie2에게는 세 번째 상영시간-의 좌석 정보를 저장하고 있다.)

3 level hierarchy

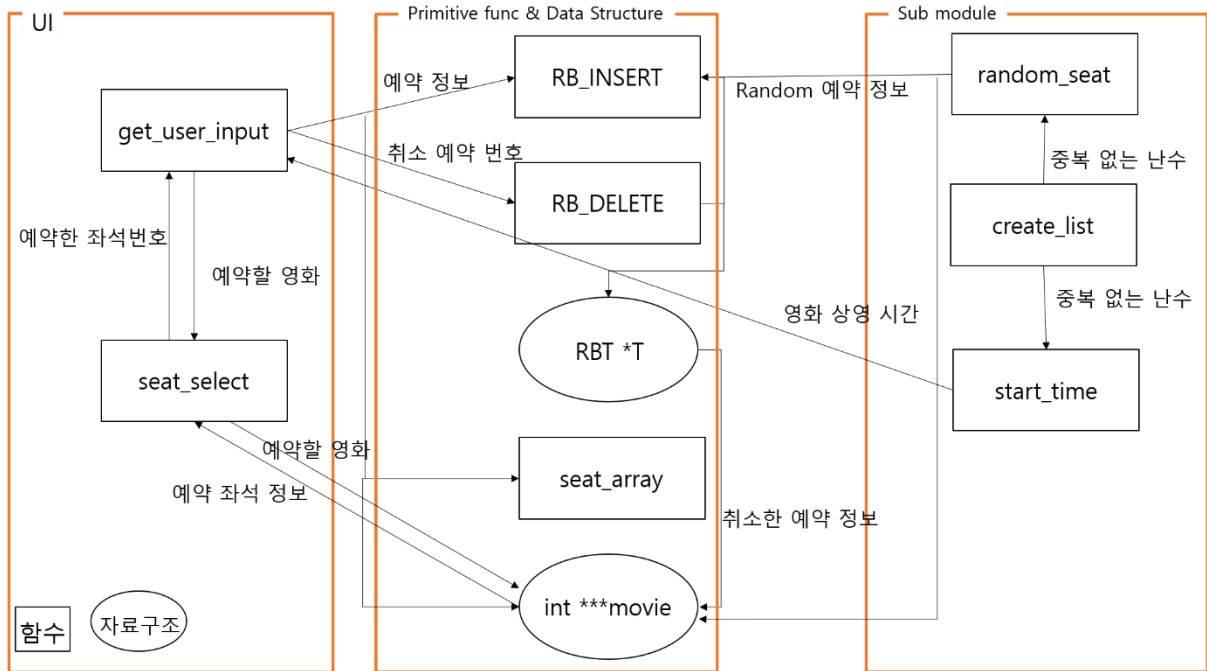


Figure 4 주요 함수와 자료구조

프로그램의 주요 함수와 자료 구조 간에 관계는 Figure 3과 같다.

1. 예약 정보
사용자로부터 입력 받은 예약 정보이다.
2. 취소 예약 정보
사용자로부터 입력 받은 취소할 예약 번호이다.
3. 예약할 영화
사용자로부터 입력 받은 예약할 영화의 번호로 `seat_select`에서는 그 영화에 해당하는 좌석의 예약 상황을 보여주고 예약할 좌석의 번호를 입력받는다.
4. 예약한 좌석번호
`seat_select`에서 반환하는 사용자가 예약한 좌석 번호다.
5. 예약 좌석 정보
예약된 좌석은 1, 예약되지 않은 좌석의 index에는 0이 저장되어 있다. 좌석은 0 ~99의 번호를 가지며 2차원 배열에서 십의 자리 수는 row, 일의 자리 수는 column이다.
6. random 예약 정보
사용자로부터 예약을 받기 전에 각 영화마다 전체 좌석의 30%(30석)를 random으로 예약한다.

7. 중복 없는 난수

중복 없이 30석을 예약하기 위한 난수, 임의의 상영 시작 시간을 위한 난수이다.

8. 영화 상영 시간

한 영화의 상영 시간은 그 날 동안은 겹치지 않는 세 정수여야 하므로 중복 없는 3개의 난수를 만든다. 범위는 0시 ~ 23시이다. 상영 시간은 편의상 정시로 정한다.

9. 취소한 예약 정보

취소한 예약의 자리를 예약되지 않은 상태로 바꾸기 위해 취소한 예약 정보를 전달한다.

세부적인 구현

1. main

- a) srand()로 실행할 때마다 새로운 난수를 생성하도록 한다.
- b) 3차원 배열 int ***movie를 정의한다.
- c) RBT T를 정의한다.
- d) 영화 상영 시간 show_time을 start_time()을 이용하여 난수로 받아 저장한다.
- e) random_seat()을 이용하여 좌석의 30%를 채운다.
- f) get_user_input()을 이용하여 사용자로부터 keyboard 입력을 받는다.
- g) 동적 할당 메모리를 반납한다.
- h) 정상적으로 종료되면 0을 return하고 프로그램 구동 중 동적 메모리 할당에 실패하면 -1을 return한다.

2. make_reserve_id

- a) 0 ~ ID_MAX - 1 (define을 이용하여 10000으로 설정해 놓았다. 예약 번호의 범위를 바꾸고 싶다면 바꿀 수 있다.) 까지의 난수를 생성한다.
- b) 만약 그 값이 이미 트리 안에 key로 존재하면 존재하지 않는 값이 나올 때까지 계속 난수를 생성한다.
- c) 생성된 난수를 return한다.

3. movie_name

- a) 영화의 번호를 입력 받으면 char로 변환해준다.

4. seat_array

- a) 한 영화의 좌석을 표현하는 2차원 배열 중 입력 받은 좌석에 이미 예약됨 혹은 예약 가능을 표현한다.

- b) 만약 reserve가 1이라면 예약된 좌석을 표시하고 0이라면 취소된 좌석을 표현한다.

5. random_seat

- a) RAND_INPUT만큼의 좌석을 각 영화마다 예매한다.
- b) RAND_INPUT은 define으로 지정되어 있는 수로 30으로 설정되어 있고 필요에 따라 변경이 가능하다.
- c) 27개의 영화에 대해서(27개의 영화는 상영 날짜, 상영 시간이 다른 영화이다. 따라서 추가로 상영 날짜, 상영 시간을 난수로 결정할 필요 없이 모든 영화에 대해 RAND_INPUT만큼의 좌석을 예약하기만 하면 된다.) 각 RAND_INPUT개의 예약을 예약 번호를 make_reserve_id를 이용해 만들고 create_list로 만든 RAND_INPUT개의 겹치지 않는 난수를 좌석 번호로 이용한다.
- d) 정해진 예약 정보를 RB_INSERT를 이용하여 트리에 추가한다.
- e) 3차원 배열 movie에도 seat_array를 이용하여 좌석 예매 정보를 저장한다.

6. create_list

- a) 입력받은 int 배열에 0 ~ range - 1까지의 난수를 size개만큼 채워 넣는다.
- b) 이 때 난수는 겹치지 않는다.
- c) linear search를 통해 만들어진 난수가 겹치면 난수를 다시 생성한다.

7. start_time

- a) int 배열 show_time에 영화의 시작시간을 뜻하는 난수를 생성하여 저장한다.
- b) create_list로 0 ~ 23까지의 난수를 3개 만들어 저장하는 것을 9번 반복한다. (날짜별로 상영시간을 무작위로 만든다.)

8. get_user_input

- a) mode를 입력 받는다. (1 예약, 2 취소, 0 프로그램 종료)
- b) mode에 따라 다음 활동을 결정한다.
- c) 예약
 - 1) 1이면 날짜를 입력 받고 무작위로 생성한 show_time을 출력하여 9가지 경우의 수 중 하나를 선택받는다.
 - 2) 영화를 선택한 후 seat_select를 이용하여 좌석 번호를 입력 받는다.
 - 3) 모든 입력은 범위를 벗어나면 올바른 범위를 입력할 때까지 다시

입력 받는다.

- 4) `seat_array`에 좌석 현황을 입력한다.
 - 5) `make_reserve_id`를 이용하여 예약 번호를 만들고 `RB_INSERT`를 이용해 예약 정보를 트리에 저장한다.
 - 6) 트리의 높이, 노드 개수, `root`와 `leaves`의 키 값을 출력한다.
- d) 취소
- 1) 예약 번호를 입력 받는다.
 - 2) 예약 번호가 트리 내에 존재하는지 `rb_search`를 이용하여 확인하고 만약 존재한다면 해당 `node`의 주솟값을 반환 받는다.
 - 3) 만약 존재하지 않는다면(반환값이 `NULL`이라면) 존재하는 예약 번호를 입력할 때까지 입력을 받는다.
 - 4) 존재하는 예약 번호에 대해 예약 번호, 상영 날짜, 상영 시간, 영화 이름, 좌석 번호를 출력한다.
 - 5) 해당 예약 번호를 `RB_DELETE`를 이용하여 트리에서 제거한다.
 - 6) `seat_array`를 이용하여 `movie`의 좌석 현황을 변경한다.
 - 7) 트리의 높이, 노드 개수, `root`와 `leaves`의 키 값을 출력한다.
- e) 프로그램 종료
- 1) `while` loop를 종료한다. – `get_user_input` 함수에서 빠져나온다.
9. `seat_select`
- a) `int **movie`를 입력 받아 해당 영화의 좌석 현황을 출력한다.
 - b) 총 100석 중 예약된 좌석 수와 예약 가능한 좌석 수를 표시한다.
 - c) 사용자로부터 좌석 번호를 입력 받는다.
 - d) 좌석 번호가 0 ~ 99 범위를 벗어나면 다시 입력 받는다.
 - e) 이미 예약된 좌석을 선택하면(2차원 배열의 해당 좌석의 값이 1이면) 다시 입력 받는다.
 - f) 유효한 좌석 번호를 반환한다.
10. `print_info`
- a) 날짜를 선택하고 나면 해당 날짜의 영화와 상영 시간을 3 x 3 표로 표현한다.
 - b) 상영 시간은 `start_time`으로 만들어진 `int` 배열 `show_time`에 저장되어 있다.

나머지 함수는 앞서 [Body.자료구조](#)에서 설명하였다.

Input

```

Press 1(for booking) 2(for canceling) 0(for termination)
1
Choose date(1 ~ 3): 1
Choose movie and time
      Movie
time      A          B          C
      1. 12:00      4. 10:00      7. 10:00
      2. 4:00       5. 18:00      8. 16:00
      3. 3:00       6. 16:00      9. 7:00
Choose the movie(1 ~ 9): 1
seat no. (can reserve) X (already reserved)
X      1      2      3      X      X      X      7      8      X
10     X      12     13     14     15     16     17     X      19
X     21     X     23     24     25     X     27     28     29
30     X     32     33     34     35     36     37     38     39
X     X     42     43     44     45     46     47     X     49
50     51     X     53     X     55     X     X     58     59
60     61     62     63     X     65     66     67     68     69
70     71     X     73     74     X     76     X     78     79
X     X     82     83     X     85     86     87     X     X
90     X     92     X     94     95     X     97     98     99
reserved seat: 30/100 available: 70/100
Choose seat no.: 12

```

Figure 5 input(예매)

Figure 5에서 주황색 네모를 표시한 부분이 사용자가 입력한 값이다.

1. 예약, 취소, 프로그램 종료 중 선택한다.
2. 날짜를 선택한다.
3. 원하는 영화와 시간을 선택한다.
4. 원하는 좌석을 선택한다.

```

Press 1(for booking) 2(for canceling) 0(for termination)
2
Enter reservation ID: 5797

```

Figure 6 input(취소)

취소를 선택하면 예약 번호를 입력할 수 있다.

Output

```

Height: 12 Node no.: 811
Root: 3108
leaf: 0 (b)23 (r)36 (r)64 (r)74 (b)94 (r)130 (r)137 (r)160 (r)178 (b)196 (r)233 (r)267 (r)294 (r)302 (r)309 (r)334 (r)38
0 (r)389 (r)401 (b)441 (r)446 (r)512 (r)539 (b)559 (r)581 (r)591 (b)614 (r)638 (r)709 (b)723 (r)774 (r)791 (r)806 (r)834
(r)846 (r)866 (b)883 (b)888 (r)909 (r)916 (r)925 (b)947 (r)965 (r)983 (b)988 (r)1019 (r)1035 (r)1039 (r)1043 (r)1097 (r
)1101 (r)1112 (r)1131 (r)1147 (r)1170 (r)1186 (r)1205 (b)1230 (r)1234 (r)1267 (r)1293 (b)1317 (b)1322 (r)1333 (r)1352 (r
)1359 (r)1375 (r)1392 (r)1398 (r)1416 (r)1431 (r)1451 (r)1463 (r)1484 (r)1527 (b)1549 (b)1553 (r)1564 (r)1600 (r)1618 (r
)1661 (r)1674 (r)1705 (r)1736 (r)1801 (r)1813 (r)1824 (r)1832 (r)1845 (r)1859 (b)1877 (r)1888 (b)1944 (r)1978 (r)1994 (r
)2015 (r)2050 (r)2060 (r)2079 (r)2094 (r)2103 (r)2125 (r)2182 (r)2237 (r)2240 (r)2258 (r)2281 (b)2305 (b)2324 (r)2349 (r
)2372 (r)2401 (r)2406 (b)2429 (r)2469 (r)2478 (r)2526 (b)2540 (r)2562 (b)2579 (b)2586 (b)2600 (r)2612 (r)2625 (r
)2663 (r)2704 (r)2728 (r)2738 (r)2744 (r)2787 (b)2832 (b)2857 (r)2988 (r)2997 (r)3064 (r)3077 (r)3081 (r)3106 (r)3123 (r
)3142 (b)3155 (r)3203 (r)3243 (b)3256 (r)3310 (r)3404 (r)3418 (b)3432 (r)3448 (r)3471 (r)3479 (r)3519 (r)3529 (r)3547 (b
)3596 (r)3618 (b)3639 (r)3652 (r)3711 (r)3758 (b)3769 (r)3782 (r)3804 (b)3808 (b)3824 (r)3837 (r)3889 (b)3914 (b)3921 (b
)3950 (r)3970 (r)3989 (r)4117 (r)4155 (r)4175 (b)4232 (b)4257 (b)4297 (b)4367 (r)4373 (r)4399 (r)4443 (b)4487 (r)4526 (r
)4548 (r)4609 (b)4617 (r)4676 (r)4702 (r)4725 (b)4748 (b)4770 (r)4809 (r)4819 (b)4862 (r)4966 (b)4984 (r)4997 (b)5003 (r
)5042 (r)5086 (r)5132 (r)5146 (r)5176 (b)5196 (r)5202 (b)5245 (r)5252 (r)5297 (r)5313 (r)5350 (r)5356 (r)5387 (r)5461 (r
)5514 (r)5566 (r)5572 (b)5609 (r)5632 (r)5698 (r)5734 (b)5761 (r)5797 (r)5800 (r)5815 (r)5831 (b)5854 (r)5878 (r)5887 (r
)5890 (r)5929 (b)5940 (r)5951 (r)5969 (r)6015 (r)6025 (b)6073 (r)6106 (b)6133 (b)6191 (b)6212 (b)6238 (r)6246 (r)6301 (b
)6386 (b)6408 (b)6463 (r)6481 (r)6494 (r)6531 (r)6547 (r)6618 (r)6636 (r)6644 (r)6652 (r)6678 (r)6728 (r)6770 (r)6795 (r
)6834 (b)6886 (r)6917 (r)6940 (b)6974 (r)7048 (r)7093 (r)7110 (r)7149 (r)7189 (r)7205 (r)7243 (r)7268 (b)7281 (b)7296 (b
)7331 (r)7424 (r)7439 (r)7445 (b)7483 (b)7498 (r)7523 (r)7544 (b)7574 (b)7585 (r)7640 (r)7676 (r)7709 (b)7741 (r)7772 (r
)7814 (r)7847 (b)7886 (b)7931 (b)7989 (r)8078 (b)8102 (b)8133 (r)8208 (r)8230 (r)8296 (r)8346 (b)8410 (r)8422 (r)8485 (r
)8515 (b)8560 (r)8638 (b)8653 (b)8734 (r)8799 (r)8802 (r)8838 (r)8909 (r)8986 (r)9008 (r)9043 (r)9056 (b)9066 (r)9094 (b
)9100 (b)9160 (b)9188 (r)9209 (r)9327 (r)9339 (r)9396 (r)9404 (b)9417 (r)9474 (r)9489 (r)9565 (r)9585 (b)9609 (b)9636 (b
)9657 (r)9686 (r)9723 (b)9739 (r)9777 (r)9787 (r)9802 (r)9823 (r)9909 (b)9950 (r)9981 (r)9999 (b)
Reservation Completed
Reservation ID: 5797
Date: 1 Movie: A
Time 12:00 Seat 12

```

Figure 7 예매 output

Figure 5과 같이 유효한 입력을 했을 경우 트리의 정보가 나오고 예매가 완료되었음을 알림과 동시에 예매 번호, 상영 날짜, 영화 이름, 상영 시간과 좌석 번호가 나온다.

```

Cancel Completed
Reservation ID: 5797Date: 1 Movie: A
Time 12:00 Seat 12
Height: 12 Node no.: 810
Root: 3108
leaf: 0 (b)23 (r)36 (r)64 (r)74 (b)94 (r)130 (r)137 (r)160 (r)178 (b)196 (r)233 (r)267 (r)294 (r)302 (r)309 (r)334 (r)38
0 (r)389 (r)401 (b)441 (r)446 (r)512 (r)539 (b)559 (r)581 (r)591 (b)614 (r)638 (r)709 (b)723 (r)774 (r)791 (r)806 (r)834
(r)846 (r)866 (b)883 (b)888 (r)909 (r)916 (r)925 (b)947 (r)965 (r)983 (b)988 (r)1019 (r)1035 (r)1039 (r)1043 (r)1097 (r
)1101 (r)1112 (r)1131 (r)1147 (r)1170 (r)1186 (r)1205 (b)1230 (r)1234 (r)1267 (r)1293 (b)1317 (b)1322 (r)1333 (r)1352 (r
)1359 (r)1375 (r)1392 (r)1398 (r)1416 (r)1431 (r)1451 (r)1463 (r)1484 (r)1527 (b)1549 (b)1553 (r)1564 (r)1600 (r)1618 (r
)1661 (r)1674 (r)1705 (r)1736 (r)1801 (r)1813 (r)1824 (r)1832 (r)1845 (r)1859 (b)1877 (r)1888 (b)1944 (r)1978 (r)1994 (r
)2015 (r)2050 (r)2060 (r)2079 (r)2094 (r)2103 (r)2125 (r)2182 (r)2237 (r)2240 (r)2258 (r)2281 (b)2305 (b)2324 (r)2349 (r
)2372 (r)2401 (r)2406 (b)2429 (r)2469 (r)2478 (r)2526 (b)2540 (r)2562 (b)2579 (b)2586 (b)2600 (r)2612 (r)2625 (r
)2663 (r)2704 (r)2728 (r)2738 (r)2744 (r)2787 (b)2832 (b)2857 (r)2988 (r)2997 (r)3064 (r)3077 (r)3081 (r)3106 (r)3123 (r
)3142 (b)3155 (r)3203 (r)3243 (b)3256 (r)3310 (r)3404 (r)3418 (b)3432 (r)3448 (r)3471 (r)3479 (r)3519 (r)3529 (r)3547 (b
)3596 (r)3618 (b)3639 (r)3652 (r)3711 (r)3758 (b)3769 (r)3782 (r)3804 (b)3808 (b)3824 (r)3837 (r)3889 (b)3914 (b)3921 (b
)3950 (r)3970 (r)3989 (r)4117 (r)4155 (r)4175 (b)4232 (b)4257 (b)4297 (b)4367 (r)4373 (r)4399 (r)4443 (b)4487 (r)4526 (r
)4548 (r)4609 (b)4617 (r)4676 (r)4702 (r)4725 (b)4748 (b)4770 (r)4809 (r)4819 (b)4862 (r)4966 (b)4984 (r)4997 (b)5003 (r
)5042 (r)5086 (r)5132 (r)5146 (r)5176 (b)5196 (r)5202 (b)5245 (r)5252 (r)5297 (r)5313 (r)5350 (r)5356 (r)5387 (r)5461 (r
)5514 (r)5566 (r)5572 (b)5609 (r)5632 (r)5698 (r)5734 (b)5761 (r)5797 (r)5800 (r)5815 (r)5831 (b)5854 (r)5878 (r)5887 (r
)5890 (r)5929 (b)5940 (r)5951 (r)5969 (r)6015 (r)6025 (b)6073 (r)6106 (b)6133 (b)6191 (b)6212 (b)6238 (r)6246 (r)6301 (b
)6386 (b)6408 (b)6463 (r)6481 (r)6494 (r)6531 (r)6547 (r)6618 (r)6636 (r)6644 (r)6652 (r)6678 (r)6728 (r)6770 (r)6795 (r
)6834 (b)6886 (r)6917 (r)6940 (b)6974 (r)7048 (r)7093 (r)7110 (r)7149 (r)7189 (r)7205 (r)7243 (r)7268 (b)7281 (b)7296 (b
)7331 (r)7424 (r)7439 (r)7445 (b)7483 (b)7498 (r)7523 (r)7544 (b)7574 (b)7585 (r)7640 (r)7676 (r)7709 (b)7741 (r)7772 (r
)7814 (r)7847 (b)7886 (b)7931 (b)7989 (r)8078 (b)8102 (b)8133 (r)8208 (r)8230 (r)8296 (r)8346 (b)8410 (r)8422 (r)8485 (r
)8515 (b)8560 (r)8638 (b)8653 (b)8734 (r)8799 (r)8802 (r)8838 (r)8909 (r)8986 (r)9008 (r)9043 (r)9056 (b)9066 (r)9094 (b
)9100 (b)9160 (b)9188 (r)9209 (r)9327 (r)9339 (r)9396 (r)9404 (b)9417 (r)9474 (r)9489 (r)9565 (r)9585 (b)9609 (b)9636 (b
)9657 (r)9686 (r)9723 (b)9739 (r)9777 (r)9787 (r)9802 (r)9823 (r)9909 (b)9950 (r)9981 (r)9999 (b)

```

Figure 8 취소 output

Figure 6처럼 적절한 예약 번호를 입력하면 취소된 예약의 정보를 출력하고 트리의 정보를 출력한다.

2. 결과

예매 & 취소

다음과 같이 테스트를 진행했다.

1. date 1의 영화 1 ~ 5까지는 예약을 진행하고 다시 예약 모드로 가서 그 좌석이 예약처리(즉X가 되었는지 확인한다.)가 되었는지 확인한다. 후에 예약한 예약 번호를 입력하여 예약 내역을 지우고 다시 예약 모드로 들어가서 그 좌석이 다시 예약 가능한 상태가 되었는지 확인한다. 따라서 결과 캡처 사진에서는 노드의 수가 합리적이지 않게 보일 수 있다. (확인을 위해 예약 모드에 여러 번 진입하므로 노드의 수가 이론적인 경우보다 크다.)

```
Press 1(for booking) 2(for canceling) 0(for termination)
1
Choose date(1 ~ 3): 1
Choose movie and time
Movie
A B C
time
1. 16:00 4. 11:00 7. 1:00
2. 14:00 5. 12:00 8. 4:00
3. 0:00 6. 0:00 9. 2:00
Choose the movie(1 ~ 9): 1
seat no. (can reserve) X (already reserved)
0 X 2 3 X 6 7 8 X
10 X 11 12 13 14 15 16 17 18 X
X X 22 23 24 25 26 27 X X
30 X 31 32 33 X 35 36 37 38 39
40 X 41 X 43 44 X 46 47 X 49
X X 51 X 53 54 X X 57 58 59
60 X 62 63 64 65 66 X X 69
70 X 72 73 74 75 X 77 X 79
X 81 82 X 84 85 X 87 88 89
90 X X X 94 95 96 X 98 99
reserved seat: 30/100 available: 70/100
Choose seat no.: 0
Height: 12 Node no.: 811
Root: 4641
leaf: 5 (b)15 (r)50 (r)64 (b)74 (r)87 (r)93 (r)123 (r)157 (r)200 (r)243 (r)259 (b)292 (b)313 (b)317 (r)386 (r)391 (b)410 (b)443 (r)456 (r)474 (r)478
(r)516 (r)529 (r)547 (b)557 (b)580 (r)598 (r)614 (b)640 (r)668 (b)725 (r)788 (r)794 (r)818 (r)859 (r)900 (r)921 (r)934 (r)967 (b)987 (r)1005 (r)1020
(r)1036 (b)1051 (r)1077 (r)1083 (b)1086 (b)1098 (r)1117 (b)1172 (b)1216 (r)1226 (r)1238 (r)1248 (r)1269 (r)1317 (r)1342 (r)1374 (b)1392 (b)1418 (b)14
34 (b)1458 (r)1459 (r)1464 (r)1472 (r)1515 (r)1530 (r)1536 (r)1588 (r)1608 (r)1613 (r)1626 (r)1669 (r)1678 (b)1695 (r)1721 (r)1731 (r)1735 (r)1766 (r)1
773 (r)1798 (r)1841 (r)1862 (r)1869 (r)1885 (r)1890 (b)1908 (r)1929 (r)1958 (r)1965 (r)1972 (r)2009 (r)2018 (r)2038 (r)2045 (r)2062 (r)2074 (b)2093
(r)2137 (r)2163 (b)2207 (b)2232 (b)2249 (r)2294 (r)2319 (b)2339 (r)2368 (r)2381 (r)2408 (r)2430 (r)2448 (r)2481 (r)2496 (b)2536 (r)2558 (b)2584 (r)2
617 (r)2652 (b)2698 (r)2706 (r)2724 (b)2764 (r)2782 (r)2834 (r)2875 (r)2882 (r)2898 (b)2907 (r)2935 (r)3017 (r)3032 (r)3086 (r)3116 (b)3129 (r)3142
(r)3190 (r)3197 (r)3233 (r)3251 (r)3323 (r)3343 (r)3408 (b)3472 (b)3504 (r)3522 (r)3554 (r)3560 (r)3647 (r)3653 (r)3686 (b)3689 (r)3764 (r)3792 (b)380
4 (b)3843 (r)3858 (r)3897 (r)3908 (b)3938 (r)3961 (r)4001 (r)4035 (r)4051 (r)4083 (b)4106 (r)4148 (r)4168 (r)4205 (b)4307 (r)4368 (r)4391 (b)4437 (r)4
457 (b)4493 (b)4558 (r)4563 (r)4585 (r)4623 (b)4649 (r)4660 (r)4679 (r)4689 (r)4707 (r)4721 (r)4780 (b)4834 (r)4869 (r)4886 (r)4916 (r)4924 (b)4981
(r)5076 (r)5080 (r)5102 (r)5182 (r)5178 (r)5238 (b)5251 (b)5289 (r)5277 (r)5343 (r)5402 (b)5417 (r)5438 (r)5483 (r)5482 (r)5513 (b)5606 (r)5668 (r)56
84 (r)5702 (r)5749 (r)5796 (b)5805 (b)5815 (r)5869 (r)5896 (b)5910 (b)5946 (r)5998 (r)6072 (r)6097 (b)6117 (r)6214 (r)6224 (r)6246 (b)6266 (r)6304 (r)6338
(r)6369 (b)6395 (r)6415 (b)6427 (r)6451 (r)6467 (b)6486 (r)6510 (r)6599 (r)6621 (b)6670 (r)6714 (r)6732 (b)6780 (r)6801 (r)6843 (r)6856 (r)6878
(r)6908 (r)6919 (r)6981 (b)6945 (r)6978 (r)7029 (b)7057 (r)7075 (r)7090 (r)7108 (r)7125 (b)7162 (r)7177 (r)7204 (b)7228 (r)7243 (r)7274 (b)7305 (r)7326
(b)7419 (b)7426 (r)7452 (r)7514 (r)7538 (b)7576 (b)7618 (r)7672 (r)7683 (b)7746 (r)7769 (r)7804 (b)7814 (b)7821 (r)7823 (r)7834 (b)7883 (r)7889 (r)7949
(r)7966 (r)7977 (r)7993 (r)8011 (b)8040 (r)8078 (b)8085 (r)8113 (b)8152 (r)8175 (r)8196 (r)8242 (r)8275 (r)8288 (b)8413 (r)8416 (b)8446 (r)8467
(r)8489 (r)8546 (r)8583 (r)8620 (r)8639 (r)8657 (r)8688 (b)8701 (r)8744 (r)8775 (b)8803 (b)8821 (r)8838 (r)8877 (b)8916 (r)9028 (r)9068 (r)9093
(b)9124 (r)9143 (r)9193 (r)9205 (b)9233 (r)9295 (r)9319 (r)9354 (r)9404 (r)9469 (r)9495 (r)9573 (r)9605 (r)9616 (b)9640 (r)9659 (r)9702 (b)9772
(r)9791 (r)9861 (b)9907 (r)9957 (r)9995 (b)
Reservation Completed
Reservation ID: 1531
Date: 1 Movie: A
Time 16:00 Seat 0
```

Figure 9 Day1 movie 1 reserve (day1-1 reserve.png)

빨간 박스 안에는 날짜, 영화, 좌석을 선택하는 모습을 볼 수 있다. 초록 박스 안에는 트리의 높이, 노드 개수, 루트의 키 값, leaf의 키 값(왼쪽부터)을 확인할 수 있다. 파란 박스 안에는 예약 번호, 상영 날짜, 영화, 상영시간과 좌석이 보여진다.


```

Press 1(for booking) 2(for canceling) 0(for termination)
1
Choose date(1 ~ 3): 1
Choose movie and time
      Movie
      A           B           C
time
  1. 16:00      4. 11:00      7. 1:00
  2. 14:00      5. 12:00      8. 4:00
  3. 0:00       6. 0:00       9. 2:00
Choose the movie(1 ~ 9): 1
seat no. (can reserve) X (already reserved)
X      X      2      3      X      X      6      7      8      X
10     11     12     13     14     15     16     17     18     X
X      X      22     23     24     25     26     27     X      X
30     31     32     33     X      35     36     37     38     39
40     41     X      43     44     X      46     47     X      49
X      51     X      53     54     X      X      57     58     59
60     X      62     63     64     65     66     X      X      69
70     X      72     73     74     75     X      77     X      79
X      81     82     X      84     85     X      87     88     89
90     X      X      X      94     95     96     X      98     99
reserved seat: 31/100 available: 69/100
Choose seat no.:

```

Figure 10 Day 1 movie 1 reservation check(day1-1 reserve check.png)

Figure9에서 좌석 0의 예약을 완료한 상태로 다시 Day1 movie1의 좌석 현황을 보면 좌석 0이 예약 처리되어 더 이상 예약할 수 없다는 것을 확인할 수 있다.

```

Press 1(for booking) 2(for canceling) 0(for termination)
2
Enter reservation ID: 1531
Cancel Completed
Reservation ID: 1531 Date: 1 Movie: A
Time 16:00 Seat 0
Height: 12 Node no.: 810
Root: 4641
Leaf: 5 (b)15 (r)50 (b)64 (b)74 (r)87 (r)93 (r)123 (r)157 (r)200 (r)243 (r)259 (b)292 (b)313 (b)317 (r)386 (r)391 (b)410 (b)443 (r)456 (r)474 (r)478
(r)516 (r)529 (b)547 (b)557 (b)580 (r)598 (r)614 (b)640 (r)668 (b)725 (r)788 (r)794 (r)818 (r)859 (r)900 (r)921 (r)934 (r)967 (b)987 (r)1005 (r)1020
(r)1036 (b)1051 (r)1077 (r)1083 (b)1086 (b)1098 (r)1117 (b)1172 (b)1216 (r)1226 (r)1238 (r)1248 (r)1269 (r)1317 (r)1342 (r)1374 (b)1392 (b)1418 (b)14
34 (b)1438 (r)1459 (r)1464 (r)1472 (r)1515 (r)1530 (r)1588 (r)1608 (r)1613 (r)1626 (r)1669 (r)1678 (b)1695 (r)1721 (r)1731 (r)1735 (r)1766 (r)1773 (r
)1798 (r)1841 (r)1862 (r)1869 (r)1885 (r)1890 (b)1908 (r)1929 (r)1958 (r)1965 (r)1972 (r)2009 (r)2018 (r)2038 (r)2045 (r)2062 (r)2074 (b)2093 (r)2137 (r
)2163 (b)2207 (b)2232 (b)2249 (r)2294 (r)2319 (b)2339 (r)2368 (r)2381 (r)2409 (r)2430 (r)2448 (r)2481 (r)2496 (b)2536 (r)2558 (b)2564 (r)2617 (r)2
652 (b)2698 (r)2706 (r)2724 (r)2764 (r)2792 (r)2834 (r)2875 (r)2882 (r)2898 (b)2907 (r)2935 (r)3017 (r)3032 (r)3086 (r)3116 (b)3129 (r)3142 (r)3190 (r
)3197 (r)3233 (r)3251 (r)3323 (r)3343 (r)3408 (b)3472 (b)3504 (r)3522 (r)3554 (r)3560 (r)3647 (r)3653 (r)3686 (b)3689 (r)3764 (r)3792 (b)3804 (b)384
3 (r)3858 (r)3897 (r)3903 (b)3938 (r)3961 (r)4001 (r)4035 (r)4051 (r)4083 (b)4106 (r)4148 (r)4168 (r)4205 (b)4307 (r)4368 (r)4391 (b)4437 (r)4457 (b)
4493 (b)4558 (r)4563 (r)4585 (r)4623 (b)4649 (r)4660 (r)4679 (r)4689 (b)4707 (r)4721 (r)4780 (b)4834 (r)4869 (r)4886 (r)4916 (r)4924 (b)4981 (r)5076
(r)5080 (r)5102 (r)5152 (r)5173 (r)5233 (b)5251 (b)5269 (r)5277 (r)5343 (r)5402 (b)5417 (r)5433 (r)5453 (r)5492 (r)5513 (b)5606 (r)5663 (r)5684 (r)57
02 (r)5749 (r)5796 (b)5805 (b)5815 (r)5869 (r)5896 (b)5910 (b)5946 (r)5998 (r)6072 (r)6097 (b)6117 (r)6214 (r)6224 (r)6246 (b)6266 (r)6304 (r)6338 (b
)6369 (b)6395 (r)6415 (b)6427 (r)6451 (r)6467 (b)6486 (r)6510 (r)6599 (r)6621 (b)6670 (r)6714 (r)6732 (b)6780 (r)6801 (r)6843 (r)6856 (r)6878 (r)6908
(r)6919 (r)6931 (b)6945 (r)6978 (r)7023 (b)7057 (r)7075 (r)7090 (r)7109 (r)7125 (b)7162 (r)7177 (r)7204 (b)7228 (r)7243 (r)7274 (b)7305 (r)7326 (b)7
419 (b)7426 (r)7452 (r)7514 (r)7538 (b)7576 (b)7618 (r)7672 (r)7683 (b)7746 (r)7769 (r)7804 (b)7814 (b)7821 (r)7823 (r)7834 (b)7883 (r)7889 (r)7949 (r
)7966 (r)7977 (r)7995 (r)8011 (b)8040 (r)8073 (b)8085 (r)8113 (b)8152 (r)8175 (r)8196 (r)8242 (r)8275 (r)8288 (b)8413 (r)8416 (b)8446 (r)8467 (r)8483
(r)8546 (r)8553 (r)8620 (r)8639 (r)8657 (r)8668 (b)8701 (r)8744 (r)8775 (b)8808 (b)8821 (r)8838 (r)8887 (b)8977 (b)9016 (r)9028 (r)9068 (r)9093 (b)
9124 (r)9143 (r)9193 (r)9205 (b)9233 (r)9295 (r)9319 (r)9354 (r)9404 (r)9469 (r)9495 (r)9573 (r)9605 (r)9616 (b)9640 (r)9659 (r)9702 (b)9772 (r)9791
(r)9861 (b)9907 (r)9957 (r)9995 (b)
Press 1(for booking) 2(for canceling) 0(for termination)
1
Choose date(1 ~ 3): 1
Choose movie and time
      Movie
      A           B           C
time
  1. 16:00      4. 11:00      7. 1:00
  2. 14:00      5. 12:00      8. 4:00
  3. 0:00       6. 0:00       9. 2:00
Choose the movie(1 ~ 9): 1
seat no. (can reserve) X (already reserved)
0      X      2      3      X      X      6      7      8      X
10     11     12     13     14     15     16     17     18     X
X      X      22     23     24     25     26     27     X      X
30     31     32     33     X      35     36     37     38     39
40     41     X      43     44     X      46     47     X      49
X      51     X      53     54     X      X      57     58     59
60     X      62     63     64     65     66     X      X      69
70     X      72     73     74     75     X      77     X      79
X      81     82     X      84     85     X      87     88     89
90     X      X      X      94     95     96     X      98     99

```

Figure 11 취소 후 확인(day1-1 cancel.png)

다시 예약한 좌석을 삭제하는 모습이다. 예약 후 받았던 예약 번호 1531을 입력

하면 예약한 내용과 동일한 내용이 출력되고 예약이 취소되었다는 문구가 보인다. 이 후 다시 Day1 Movie 1의 좌석 현황을 확인하면 0번 좌석이 다시 예약 가능상태가 된 것을 알 수 있다. 즉 예약과 취소가 잘 진행되고 좌석에도 바로 반영된다는 것을 알 수 있다. 이와 같은 방식으로 Day1의 영화 5까지 test를 진행하였다.

2. 1.에서 예약이 잘 진행되고 있음을 확인했으므로 date1의 영화 6부터는 예약을 진행하고 그 예약 번호의 예약 내역을 취소하는 것만 보이도록 한다.
3. 만약 더 다양한 데이터를 보고 싶다면 직접 시뮬레이션을 진행해도 좋다.
4. 사용자가 입력하기 전 랜덤으로 예약된 좌석에 대해서도 취소를 진행할 수 있다. 취소한 예약 내역에 해당하는 영화를 선택하여 좌석 현황을 보면 그 좌석의 예약이 잘 취소되었고 실제로 예매도 가능한 상태가 된 것을 확인할 수 있다. (random seat cancel.png, random seat cancel check.png)
5. root의 key 값을 확인할 수 있으므로 root도 취소해보았다. root를 취소한 후 출력된 트리의 root key 값이 바뀐 것을 확인할 수 있다. (root delete.png)
6. 연속적인 예매를 진행하여 예매 정보가 시스템에 잘 반영되는지 확인하였다. (연속 예약1.png, 연속 예약2.png, 연속 예약3.png, 연속 예약4.png)

잘못된 입력

1. 사용자가 정해진 범위 외의 입력을 하면 적절한 범위를 알려주고 다시 입력을 받는다.

```
Choose date(1 ~ 3): 2
Choose movie and time
Movie
time      A          B          C
1. 8:00    4. 9:00    7. 20:00
2. 4:00    5. 14:00   8. 21:00
3. 21:00   6. 1:00    9. 12:00
Choose the movie(1 ~ 9): 8
seat no. (can reserve) X (already reserved)
X 1 2 X 4 X 6 X 8 9
10 11 12 13 14 X 16 17 18 19
20 21 X 23 24 25 26 27 X 29
X 31 32 33 X 35 36 37 38 39
X 41 42 43 44 45 46 X X 49
50 X 52 X 54 55 56 57 58 59
X 61 62 63 X X 66 X X 69
70 X 72 73 X 75 76 77 78 79
X X 82 83 84 85 86 87 X 89
X 91 92 X X 95 X X 98 99
reserved seat: 29/100 available: 71/100
Choose seat no.: 100
improper input. (choose 0 ~ 99)
Choose seat no.: -1
improper input. (choose 0 ~ 99)
Choose seat no.: 0
The seat is already reserved.
Choose seat no.: 1
Height: 12 Node no.: 811
```

Figure 12 범위를 벗어난 입력

```
Press 1(for booking) 2(for canceling) 0(for termination)
4
improper input.
Press 1(for booking) 2(for canceling) 0(for termination)
1
Choose date(1 ~ 3): 4
improper input (Choose 1 ~ 3)
Choose date: 1
Choose movie and time
Movie
time      A          B          C
1. 14:00   4. 2:00    7. 23:00
2. 9:00    5. 21:00   8. 20:00
3. 22:00   6. 10:00   9. 21:00
Choose the movie(1 ~ 9): 10
improper input. (choose 1 ~ 9)
Choose the movie(1 ~ 9): 1
seat no. (can reserve) X (already reserved)
X X 2 X 4 X 6 X 8 9
X 11 12 13 14 X 16 17 18 19
20 21 X 23 24 25 26 27 X 29
X 31 32 33 X 35 36 37 38 39
X 41 42 43 44 45 46 X X 49
50 X 52 X 54 55 56 57 58 59
X 61 62 63 X X 66 X X 69
70 X 72 73 X 75 76 77 78 79
X X 82 83 84 85 86 87 X 89
X 91 92 X X 95 X X 98 99
reserved seat: 31/100 available: 69/100
Choose seat no.:
```

Figure 13 범위를 벗어난 입력2

C. Conclusion

가정

1. 좌석을 10 X 10의 형태로 배정했다.
2. 좌석은 0부터 99까지 선택 가능하다.
3. 시스템을 한 번 실행한 동안에는 영화의 상영 시간이 고정된다. (random 상영 시간을 정하는 함수를 첫 한 번만 실행한다.)
4. 영화의 상영 시간은 편의 상 정시로 한다.
5. 예약 번호는 random으로 생성되며 예약 번호는 예약 내용과 무관하다.
6. 취소된 예약 번호는 다른 예약에 다시 사용될 수 있다.
7. 총 예약 개수는 10000개를 넘지 않는다. (모든 영화의 좌석이 다 꽉 차는 경우가 최대로 예약 개수가 많은 경우라 볼 수 있는데 이 때에도 $27 * 100$ 즉 2700개의 예약만 존재할 것이므로 합당한 범위라고 볼 수 있다. 만약 원한다면 코드의 define 부분에서 ID_MAX 값을 변경하면 된다.)
8. 사용자는 한 번에 한 좌석만 예약할 수 있다. (만약 여러 좌석을 예약하고 싶다면 각 좌석에 대해 예약을 따로 진행해야 한다.)
9. 사용자가 입력한 예약 번호가 만약 시스템 내에 존재한다면 추가적인 확인 절차 없이 예약을 취소한다. (즉 만약 사용자가 실수로 예약 번호를 잘못 입력하였지만 그 예약 번호가 시스템 내에 존재한다면 취소를 진행한다.)
10. 영화 상영 시간의 범위는 0시부터 23시이다. 즉 심야, 새벽 영화도 예매가 가능하다.
11. 영화관 내에 영화 상영관의 개수는 충분하여 한 영화를 여러 상영관에서 상영할 수도 있다. (즉 만약 영화 상영에 2시간이 걸리는 영화가 12시에 시작했다면 13시에 시작하는 경우도 가능하다.)
12. 30개의 임의 예약이 영화당 존재한다. 이 수는 define RAND_INPUT에서 변경 가능하다.

제한

1. 출력을 console에 모두 하는 것보단 실제로 예매에 관한 정보는 console에 출력하고 트리에 대한 정보는 output 파일을 만들어 그곳에 출력하였으면 UI가 조금 더 깔끔했을 것 같다.
2. 예약 취소를 효율적으로 하기 위해 한 트리에 모든 예약 정보를 저장하였는데 이 구조는 영화 각각의 상태를 알아보고자 할 때는 부적절한 것 같

다. 따라서 날짜별로 혹은 영화 별로 트리를 구성하는 것이 각 영화마다 트리를 만드는 것과 한 트리에 모두 저장하는 것의 절충안이 될 수 있지 않을까 예상된다.

결론

이 프로젝트를 통해 search의 시간 복잡도가 작은 Red Black Tree를 이용하는 실생활 예를 구현해볼 수 있었다. 비록 콘솔에 트리의 정보도 출력해서 UI가 미흡하다는 점, 한 트리에 예약 정보를 모두 저장하여 예약을 취소하거나 해당 예약 번호의 예약 내용을 찾는 것은 수월하나 영화당, 혹은 날짜 당 정보를 얻는 것에는 어려움이 있다는 제한점이 있으나 알고리즘 수업을 통해 배운 자료구조를 이용하여 프로젝트를 구현할 수 있게 된 것에서 큰 진전이 있어 보인다.

다른 프로젝트를 진행할 때에는 어레이, 구조체 어레이 등의 자료구조만을 사용하였는데 알고리즘 수업을 통해 필요에 따라 스택, 큐, 트리 등의 자료구조를 적절히 사용할 수 있게 되었다. 이전에는 버블 정렬만 이용하였다면 이진 트리, 이진 탐색, 퀵 정렬 등을 이용하여 효율적인 코드를 작성할 수 있게 되었다. 이를 통해 프로그램의 시간 복잡도를 줄여 후에 규모가 큰 코드를 작성할 때 실행시간을 줄일 수 있을 것이다. 또한 본인이 작성한 코드가 시간 복잡도 측면에서 효율적인지도 파악해볼 수 있을 것이다. 나아가 프로그램을 설계하는 능력이 향상되었다고 볼 수 있다.