# Algorithm Report3 2020 fall

| | | | |
|---|---|---|---|
| 설계교육계획서 | | | |
| 설계학점 | 1 | 설계기간 | 2020 학년도 2 학기 |
| 1. 설계주제 (objecctive) | Multiplex theater reservation system | | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2. 관련 설계요소 | 구 성 요 소 | | | | | 제 한 조 건 | | | | |
| | 목표 설정 | 합성 | 분석 | 제작 | 시험및 평가 | 시스템 구성제약요건 | 성능 | 안정성 | 신뢰성 | 경제성 |
| | O | | O | O | O | O | O | | O | |

| | |
|---|---|
| 3. 운용방안 | |
| 구성원 (team composition) | Individual work (due date: 5 PM, Dec. 10, 2020) |
| 수행방법<br>(work plan) | * 11th week: Selection of the target system and the designing of various submodules.<br>- Finalize the design of the target system according to requirements given below.<br>- Determine the objective of the target system including detailed functionalities (services that system provides) of the system (what the system can do, also can't do). Anything that are related to the system.<br>Minimum service requirement is to provide the reservation as shown below.<br>- Detailed design of the input and output values of the determined functionalities.<br>- Design of the submodules for the system<br>- Design of the user interface for the system.<br>* 12th ~ 13 th week: Implementation of the designed system (with at least three level's of hierarchy)<br>- Implementation of the data structures and primitive functions<br>- Implementation of the submodules<br>- Implementation of the user interface<br>- Integration of these components<br>- Testing of the submodules<br>- Testing of the integrated system using UI<br>* 14th week: Writing of the report and submission |
| 제출결과물<br>(submission) | Report and program |

## 4 구체적인설계방법 (detailed description of the design and the requirements)

▶ Basically, an open ended problem.
Unless otherwise stated, all the designs are up to the programmer.

## [Minimum requirements]

▶ Input and output description

Input: case 1) for reservation: movie id (or name), time (start time) and date, seat number

※ The information table showing above information should be displayed in the monitor once the reservation system is executed so that users can provide required inputs.

The seat layout of seat status display (vacant or occupied seat status for each movie) for each movie should also be displayed (must be visually identifiable).

case 2) for reservation cancellation: reservation id

Output: for case 1) The reservation confirmation by showing the following information => reservation id, movie id (name), time (start time) and date, seat number

for case 2) Cancellation confirmation comment including the cancelled reservation number.

▶ Constraint of the reservation system:

- 3 different movies and each with 3 show times per day
- Can reserve the movies for the next 3 days.
- 100 seats per movie

▶ Reservation table: Must be implemented by the red black tree(s) (rb-trees).
How many rb-trees should be constructed is up to the programmer.

- Reservations should be inserted into the rb-tree with proper key (reservation id) order.
- A node of the rb-tree contains all the information regarding the reservation.

▶ Other guide lines

1) Any extra structure usage, any assumption, and all other decisions related to the reservation is entirely up to you.
2) The rb-tree should make use of at least the following primitive functions; rb-insert, rb-delete, print. The print function displays the layout of seat status.
3) There should be separate UI (User Interface) module or function.
UI gets inputs from the user, and displays outputs of the reservation confirmation and the layout of seat status (UI does not mean graphic user interface, it can be just text based interface).

▶ RB tree data structure:

- Initially the seats for all the movies should be 30% filled by randomly generated reservations (reservation id, movie id, show time and date for a movie, seat number,).
- Range of values should be reasonable. (eg. 26:00 is out of range)
- Print the total number of the nodes, the key values of root and leaves (from left to right), and height of the tree that is being modified each time the rb-insert or rb-delete is executed.

5. Content of the document file: the reservation system description

1) Introduction

What is the purpose of the system, brief input and out descriptions, how the rb-trees are constructed, what are the additional data structures used, important functions description, how random values are generated and range of values, etc.

2) User interface description:

What is the exact input values in order to execute the program.
What are the exact outputs.
What is displayed during the program execution other than the inputs and outputs.

3) Conclusions

Anything that you want to discuss or explain regarding the reservation system including all the assumptions and restrictions.

6. 평가방법 (Evaluation)

Evaluation will be done based on the following categories

1) Implementation quality (50%) - closeness to the program requirements
2) Diversity of the execution (testing) examples (30%)
eg) Reservations on all movies for all 3 days (at least 1 reservation per movie per time per day).
Cancellations from all movies for all 3 days (at least 1 cancellation per movie per day).
3) Modularity and comments within the program (10%)
4) The document file quality (The reservation system description) quality (10%)
5) Implementing the reservation system that does a single day reservation instead of three days reservation, every requirement is same as before, will get 95% of the total points.

## How to submit the report.

▶ Need to upload in the i-campus a zip file for the report3.
Refer to the manual file for uploading in the i-campus.

▶ The zip file contains the following three files.
1) Document file (.hwp, photo, or scan): The reservation system description
2) C program file(s): Program source code.
3) Test result file(s): Contains all the screen copy of the test results.

▶ The zip file should be named as shown below,

report3_id_name.zip
example) report3_2020123456_HongGilDong.zip

The zip file contains above 1), 2), and 3).