Algorithm Report3 2022 fall

<table>
<tr><td colspan="11">설계교육계획서</td></tr>
<tr><td>설계학점</td><td colspan="4">1</td><td>설계기간</td><td colspan="5">2022 학년도 2 학기</td></tr>
<tr><td>1. 설계주제
(objecctive)</td><td colspan="10">Airline reservation system</td></tr>
<tr><td rowspan="3">2. 관련 설계요소</td><td colspan="5">구 성 요 소</td><td colspan="5">제 한 조 건</td></tr>
<tr><td>목표
설정</td><td>합성</td><td>분석</td><td>제작</td><td>시험및
평가</td><td>시스템
구성제
약요건</td><td>성능</td><td>안정성</td><td>신뢰성</td><td>경제성</td></tr>
<tr><td>O</td><td></td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td></td><td>O</td><td></td></tr>
<tr><td colspan="11">3. 운용방안</td></tr>
<tr><td>구성원 (team
composition)</td><td colspan="10">Individual work (due date: 5 PM, Nov. 25, 2022)</td></tr>
<tr><td>수행방법
(work plan)</td><td colspan="10">Design and Implementation of an airline reservation system

▶ 11th week: Selection of the target system and the design of various submodules.

- Finalize the design of the target system according to requirements given below.

- 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.

Minimum service requirement is to provide the functions as shown below.

- Detailed design of the input and output values of the determined functionalities.

- Design of the submodules for the system

- Design of the user interface for the system.

▶ 12th ~ 13 th week: Implementation of the designed system
(with at least three level's of hierarchy)

- level1: Implementation of the data structures and primitive functions
- level2: Implementation of the submodules
- level3: Implementation of the user interface
- Integration of these components
- Testing of the submodules
- Testing of the integrated system using UI

▶ 14th week: Writing of the report and submission</td></tr>
<tr><td>제출결과물
(submission)</td><td colspan="10">Report and program</td></tr>
</table>

4 구체적인설계방법 (detailed description of the design and requirements)

▶ Basically, an open ended problem.
Unless otherwise stated, all the designs are up to the programmer as long as the purpose of the system is fulfilled.

[Minimum requirements]

▶ Input and output description

Input: name, source, destination, date.

example) name1, y, g, 22

Output: name, reservation number (or id), flight schedule with flight path.

example 1) name1, 1, y(10:30am, 22)-b(2:00pm, 22)-g(9:00pm, 22)

y(departure time at source) – b(departure time at intermediate city) – g(arrival time at destination)

example 2) name2, 2, h, k, h(9:00am, 5)-a(11:00am, 5)-z(9:00pm, 5)-k(1:00am, 6)

- flight path can be any path (first path found during the graph search)
- flight path should include the intermediate cities in case of indirect path
- Assume there is only one time zone.

▶ Reservation record: red black tree (rb-tree)

- Reservations should be inserted into the rb-tree with proper key (reservation number or id) order.
- A node contains all the necessary information regarding the reservation.
- A flight path from a source to a destination can be either the direct path or indirect path (there can be no path between any two cities).
- A flight time between any two cities is determined by the locations. Assume the airplane speed is 500 km/hour.
- The reservation record (rb-tree) is initially empty.

▶ Flight path: Graph data structure with adjacency list

- Shows bidirectional direct paths among cities (26 cities). An edge of the graph represents bidirectional path.
- A city name is a single alphabet character (from a ~ z).
- Randomly generate 10 distinct direct paths for every city (There are direct paths from a city to 10 other cities).
  ※ If a path already exists, ignore this and generate new one. ※ There may be no path between any two cities.
- A city location is randomly generated and represented by (x, y),
  where 0 kms <= x <=W3000 km or E3000 km, 0 kms <= y <=N3000 km or S3000 km,
- The graph consists of edges with distance (unit is km).

▶ Departure time table for one month (1^{st} ~ 31^{st}): Select a data structure of your choice.

- For all the cities, departure times for all 10 paths are randomly generated everyday for one month. (Need to write a function for this.)
- The table contains one month schedule of all the cities departure times.

▶ Execution:

Insertions and deletions of reservations (at least 10 insertions with an actual path and 5 deletions from the rb-tree, deletion by the key or id) should be executed for testing the system.

Insertion test: print the rb-tree before & after each insertion

Deletion test: print the rb-tree before & after each deletion

※ print with rb-print function.

▶ Other restrictions, assumptions and additional functionalities should be described if there is any.

▶ Other guide lines

1) Any extra structures usage and all other decisions related to the system are entirely up to you.
2) The rb-tree should make use of at least the following primitive functions: rb-insert, rb-delete, rb-print. Where, rb-print prints with the key (id) and color info of all the nodes in rb-tree.

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

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 system including all the assumptions and restrictions.

6. 평가방법 (Evaluation)

Evaluation is done based on the following categories

1) Program quality (60%) - closeness to the program requirements and correctness execution
2) Diversity of the execution (testing) examples (20%)
3) Modularity and comments within the program (10%)
4) The document file quality (The system description) quality (10%)

How to submit the report.

- Need to upload in the i-campus a zip file for report3.
  Refer to the manual file for uploading in the i-campus.
- The zip file contains the following three files.
  1) Document file (.hwp, .doc, photo, or scan): Explanation of the system.
  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).
- Use window OS and visual studio program.

[Simplified version of the airline reservation system for submission]

You can work on this report according to the following simplified minimum requirements.
(Rest of the minimum requirements are same as the previous non-simplified requirements.)

Simplified minimum requirements)

▶ One month schedule of all the cities departure times. => One day schedule of all the cities departure times.

▶ A flight time between any two cities is determined by the locations. =>
A flight time between any two cities is one hour.

▶ The "name" shown below can be just name1, name2, name3, ... (not arbitrary alphabets).
Input: name, source, destination, date.
Output: name, reservation number (or id), flight schedule with flight path.