VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Assignment 2 - Phase 2

URBAN BUS SYSTEM (Cont.)

HO CHI MINH CITY, 12/2020

ASSIGNMENT 2 SPECIFICATION (Phase 2)

Version 1.0

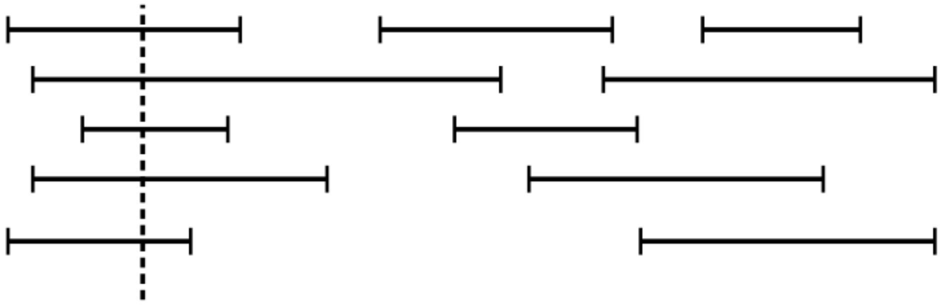
1 Outcome

After successfully completing this assignment, students will be able to:

- Apply a known self-balancing tree structure to solve real-life problems.
- Design an algorithm to accomplish real-life problems.
- Analyse the complexity of the implemented algorithm based on empirical study.

2 Introduction

At the bus stops in the bus system, there also exists some parking areas. Between two consecutive trips, the buses frequently have to wait for certain procedures. Hence, the problem for capacity of parking areas must be raised to administrators.



For example, on the above figure, each line and closed-interval demonstrate a bus and the time each bus have to wait for processing procedure prior to the next trip respectively. It can be seen that, there is a time that requires the minimum number of 5 parking slots for 5 different buses waiting for handling procedure.

3 Tasks

3.1 Task 1 (4 points)

Students are required to design and implement an algorithm using C++ to calculate the minimum parking slots that a bus station demands to accommodate all the waiting buses at the station.

- Input: List of half-closed intervals: $[s_i, t_i)$ corresponds to the arrival time to the parking area (starting to process the procedure) and the departure time for the next trip (denoted by a closed-interval in the example figure).
- Output: the minimum number of parking slots required by a bus stop.
- Complexity requirement: $O(N \log N)$ where N is the number of half-closed interval.

3.2 Task 2 (6 points)

Students are asked to construct a structure named **BusParking** using C++ to store information about the half-closed intervals so that performing query about minimum parking slots at anytime will result in the complexity of $O(1)$. Students are allowed to use standard libraries STL in C++.

The **BusParking** structure must provide 3 following methods:

- `insert(s, t)`: insert a half-closed interval $[s, t)$ into the structure, knowing that every $[s, t)$ differ from others by the value of two ends, and will not be inserted if existed. The complexity of this method is always $O(\log N)$.
- `remove(s, t)`: delete an existed half-closed interval $[s, t)$ out of the structure. The complexity of this method is always $O(\log N)$.
- `minPark()`: return the number of minimum parking slots. The complexity of this method is always $O(1)$.

4 Regulation and Submission

Students will be provided with an initialization code framework on the general course site. Students code directly on the BKeL system and the student's final submission will be used for grading.

All of the students' questions about this assignment will be answered on the course site forum. DO NOT SEND ANY EMAIL FOR QUESTIONING to the teachers or teaching assistants under any circumstances.

———————THE END———————