

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



---

Bài tập lớn 2 - Phần 2

# HỆ THỐNG XE BUS TẠI CÁC THÀNH PHỐ LỚN (TT)

---

TP. HỒ CHÍ MINH, THÁNG 12/2020

# ĐẶC TẢ BÀI TẬP LỚN 2 (Phần 2)

Phiên bản 1.0

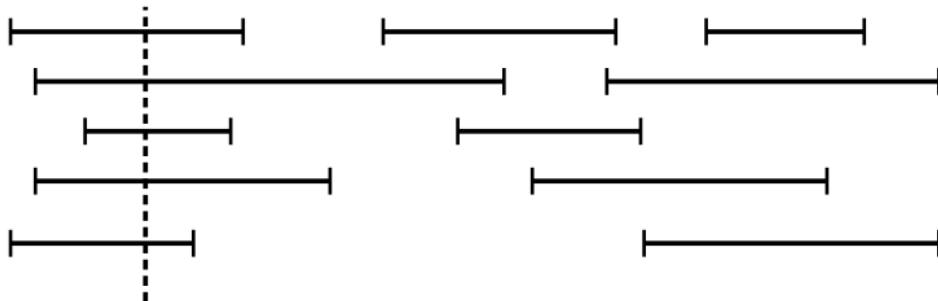
## 1 Chuẩn đầu ra

Sau khi hoàn thành bài tập lớn này, sinh viên sẽ có khả năng:

- Ứng dụng cấu trúc cây cân bằng đã biết vào xử lý bài toán thực tế.
- Thiết kế được một giải thuật để giải quyết một bài toán thực tế
- Phân tích độ phức tạp của giải thuật đã hiện thực dưới góc độ thực tế thí nghiệm.

## 2 Dẫn nhập

Hệ thống xe bus tại các trạm dừng của các chuyến xe còn có các bến đỗ xe. Giữa hai chuyến liên tiếp, các xe thường phải chờ để thực hiện một số thủ tục nhất định. Do vậy, sức chứa của các bến là vấn đề mà các ban quản lý của các bến cần phải đặt ra.



Ví dụ như trên hình vẽ mỗi dòng tương ứng với một chiếc xe. Các đoạn tương ứng cho thời gian các chuyến xe phải chờ để làm thủ tục cho chuyến tiếp theo. Như vậy, có thời điểm, bến xe cần ít nhất là 5 chỗ đậu để có thể chứa hết được 5 chiếc xe khác nhau đang chờ làm thủ tục.

## 3 Nhiệm vụ

### 3.1 Nhiệm vụ 1 (4 điểm)

Sinh viên được yêu cầu thiết kế và hiện thực một giải thuật trên ngôn ngữ C++ để có thể tính toán ra số chỗ đậu tối thiểu mà một bến xe cần phải có để có thể chứa được toàn bộ xe bus.

- Đầu vào: Danh sách các nửa đoạn:  $[s_i, t_i)$  tương ứng với thời điểm đến bến đậu (bắt đầu làm thủ tục) và thời điểm xuất phát cho chuyến tiếp theo (tương ứng với một đoạn trên hình ví dụ).
- Đầu ra: số chỗ đậu tối thiểu mà một bến xe cần phải có.
- Yêu cầu về độ phức tạp:  $O(N \log N)$  với  $N$  là số nửa đoạn.

### 3.2 Nhiệm vụ 2 (6 điểm)

Sinh viên được yêu cầu xây một cấu trúc **BusParking** trên C++ để hiện thực việc lưu trữ thông tin các nửa đoạn trên sao cho việc truy xuất chỗ đậu tối thiểu tại bất kỳ thời điểm nào đều có độ phức tạp là  $O(1)$ . Sinh viên có thể sử dụng các thư viện chuẩn STL của C++.

Cấu trúc **BusParking** phải cung cấp ra được 3 phương thức:

- `insert(s, t)`: thêm một nửa đoạn  $[s, t)$  vào trong cấu trúc. Biết rằng, các  $[s, t)$  không giống nhau ở giá trị hai đầu, nếu đã tồn tại thì không thêm vào cấu trúc. Độ phức tạp của thao tác này luôn là  $O(\log N)$ .
- `remove(s, t)`: xóa một nửa đoạn  $[s, t)$  đã tồn tại ra khỏi cấu trúc. Độ phức tạp của thao tác này luôn là  $O(\log N)$ .
- `minPark()`: trả ra số chỗ đậu ít nhất cần phải có. Độ phức tạp của thao tác này luôn là  $O(1)$ .

## 4 Quy định về thắc mắc và nộp bài

Sinh viên sẽ được cung cấp khung code khởi tạo trên site chung của môn học. Sinh viên làm bài trực tiếp trên hệ thống của BKeL và bài làm cuối cùng của sinh viên được sử dụng để chấm điểm.

Sinh viên được giải đáp thắc mắc trên diễn đàn trên site môn học, TUYỆT ĐỐI KHÔNG GỬI EMAIL cho giảng viên phụ trách để đặt câu hỏi.

—————HẾT—————