



HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.



**ĐẠI HỌC
BÁCH KHOA HÀ NỘI**
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Bài tập lớn môn cấu trúc dữ liệu và giải thuật

Đề tài : Quản lí hoạt động của sinh viên

ONE LOVE. ONE FUTURE.

1. Giới thiệu bài toán.
2. Giới thiệu thành viên.
3. Phương pháp giải quyết.
4. Kết quả thực nghiệm.
5. Kết luận.

1. Giới thiệu bài toán

Bài toán: Quản lý lịch hoạt động sinh viên

Input	Output
Thời khoá biểu sinh viên, các sự kiện kèm theo mức độ ưu tiên	Thời khoá biểu đã được cập nhật thêm các sự kiện

Động lực thực hiện: +) Tối ưu hoá quản lý thời gian

+) Ứng dụng công nghệ trong cuộc sống

+) Thúc đẩy sự sáng tạo trong lập trình

Ứng dụng: +) Môi trường giáo dục (Ehust, Ictsv,...)

+) Tổ chức cá nhân và doanh nghiệp (GG Calender, Outlook)

Thành viên nhóm

Lê Xuân Ngọc
Nhóm trưởng
20233553

Nhiệm vụ:

- Code chính
- Xây dựng thuật toán.
- Tìm hiểu và xây dựng cấu trúc dữ liệu và giải thuật của bài toán.
- Phân chia nhiệm vụ
- Tối ưu hóa thuật toán

Mức độ hoàn thành
100%

Lâm Thành Minh
Thành viên
20233526

Nhiệm vụ:

- Hỗ trợ chỉnh sửa và cải tiến code
- Tìm hiểu và lên ý tưởng CTDL và giải thuật của bài toán
- Làm slide
- Hỗ trợ xây dựng lưu đồ thuật toán

Mức độ hoàn thành
nhiệm vụ 100%

Trần Đức Đạt
Thành viên
20233311

Nhiệm vụ:

- Xây dựng bộ test
 - Tìm hiểu và lên ý tưởng cấu trúc dữ liệu và giải thuật của bài toán
 - Hỗ trợ chỉnh sửa và hậu kỳ video
- Mức độ hoàn thành
100%

Nguyễn Lâm Anh
Thành viên
20239634

Nhiệm vụ:

- Kiểm thử
- Tìm hiểu và lên ý tưởng về CTDL và giải thuật của thuật toán, xây dựng lưu đồ thuật toán
- Làm slide
- Ý tưởng phát triển và tối ưu hóa thuật toán

Mức độ hoàn thành
100%



Phương pháp giải quyết-Dữ liệu

- Ngôn ngữ lập trình: C++
- Dữ liệu:

Các loại cấu trúc dữ liệu sử dụng:

1. Struct

Struct sukien (40 bytes)

Đại diện cho một sự kiện, lưu trữ thông tin chi tiết như mô tả, ngày, giờ bắt đầu, giờ kết thúc và độ ưu tiên.

Hỗ trợ so sánh (`operator<`) để sử dụng trong các cấu trúc dữ liệu như `priority_queue`.

2. Cây AVL (AVL Tree)

< Class cay_avl > (56*n <n là số lượng sự kiện>)

Lưu trữ các sự kiện theo thứ tự giờ bắt đầu, giúp cân bằng và tối ưu hóa việc tìm kiếm, thêm, và duyệt sự kiện,

struct node: Một nút trong cây AVL, chứa thông tin về một sự kiện (`sukien`), các con trái/phải và chiều cao của nút.

Phương pháp giải quyết-Dữ liệu

3. Hàng đợi (Queue)

Class hangdoi:

- Lưu trữ các sự kiện bị xung đột (trùng lặp thời gian) để xử lý tuần tự.

`queue<sukien>` (STL Queue)

Dùng hàng đợi chuẩn từ thư viện STL để quản lý danh sách sự kiện xung đột.

* Hàng đợi ưu tiên: `Priority_queue`: Sắp xếp các hàm theo một chuẩn ưu tiên mà không cần dùng `sort`.

4. Mảng động (vecto) ($n * \text{sizeof}(\text{event})$)

Vecto 2 chiều: `vector<vector<sukien>>` tkb` trong lớp `thoikhoabieu`: Đại diện cho thời khoá biểu, lưu trữ danh sách các sự kiện theo từng ngày trong tuần (từ thứ 2 đến thứ 7)

Vecto 1 chiều: `vector<sukien>` `sukienmoilist` `:.

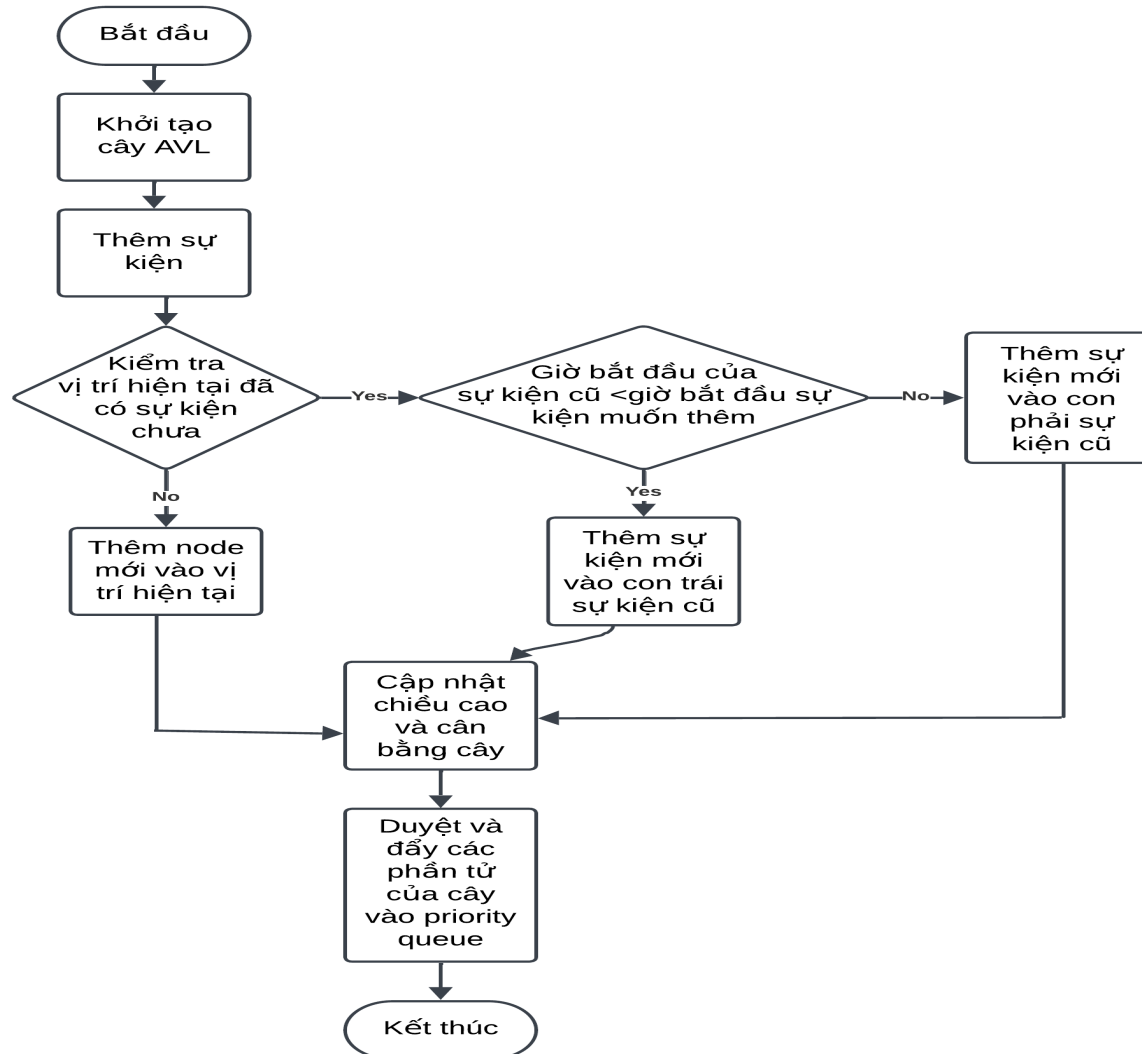
- Lưu trữ sự kiện được nhập từ người dung, kiểm tra xung đột khi thêm sự kiện và hiển thị tkb

Phương pháp giải quyết-giải thuật

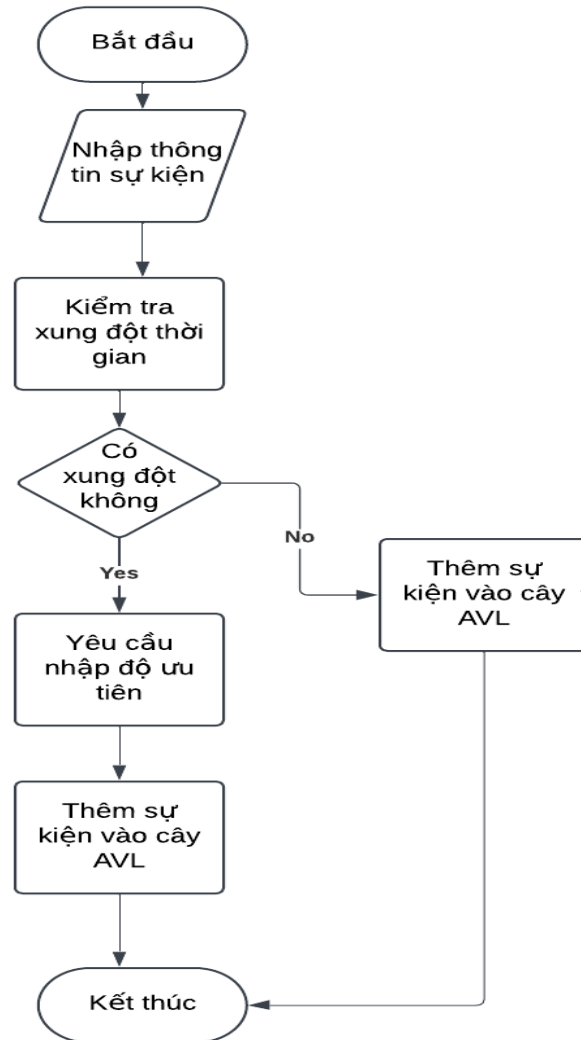
Chia bài toán thành những bài toán nhỏ và xây dựng cấu trúc dữ liệu và giải thuật từng bài toán(sử dụng lưu đồ thuật toán)

1. Thêm sự kiện.
2. Hiển thị sự kiện.
3. Quản lý xung đột .
4. Cân bằng cây AVL và đưa vào priority queue

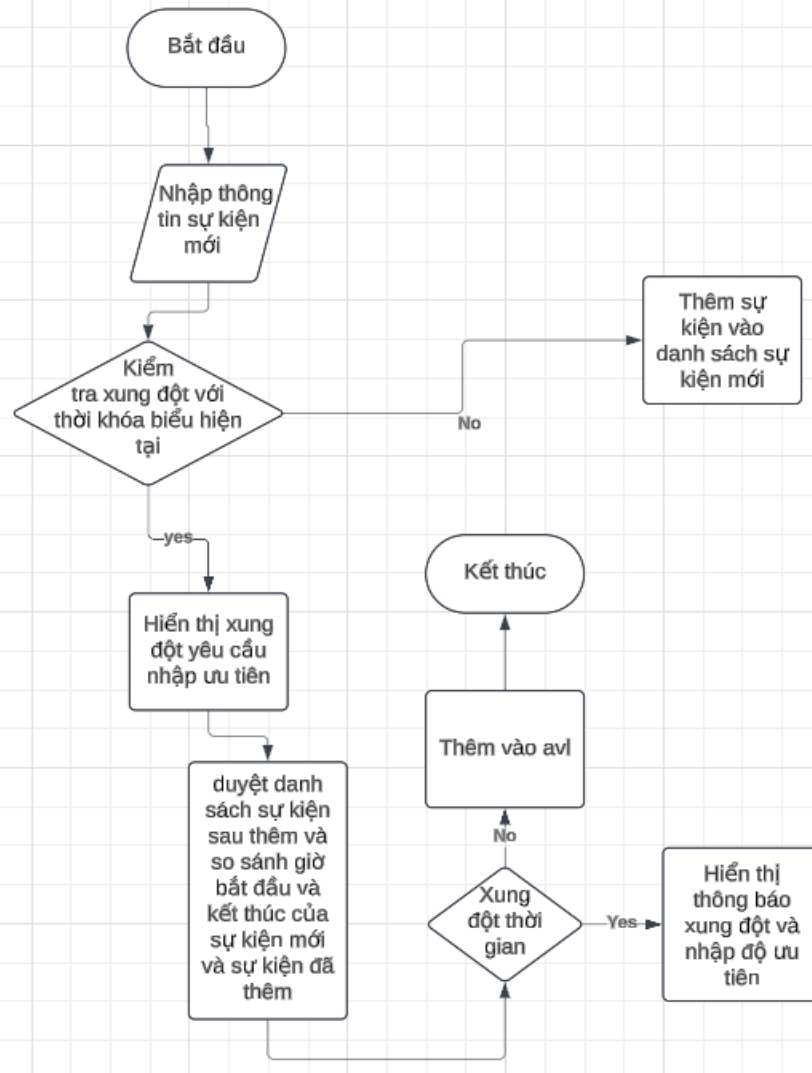
TẠO CÂY AVL VÀ DUYỆT CÂY TÌM SỰ KIỆN



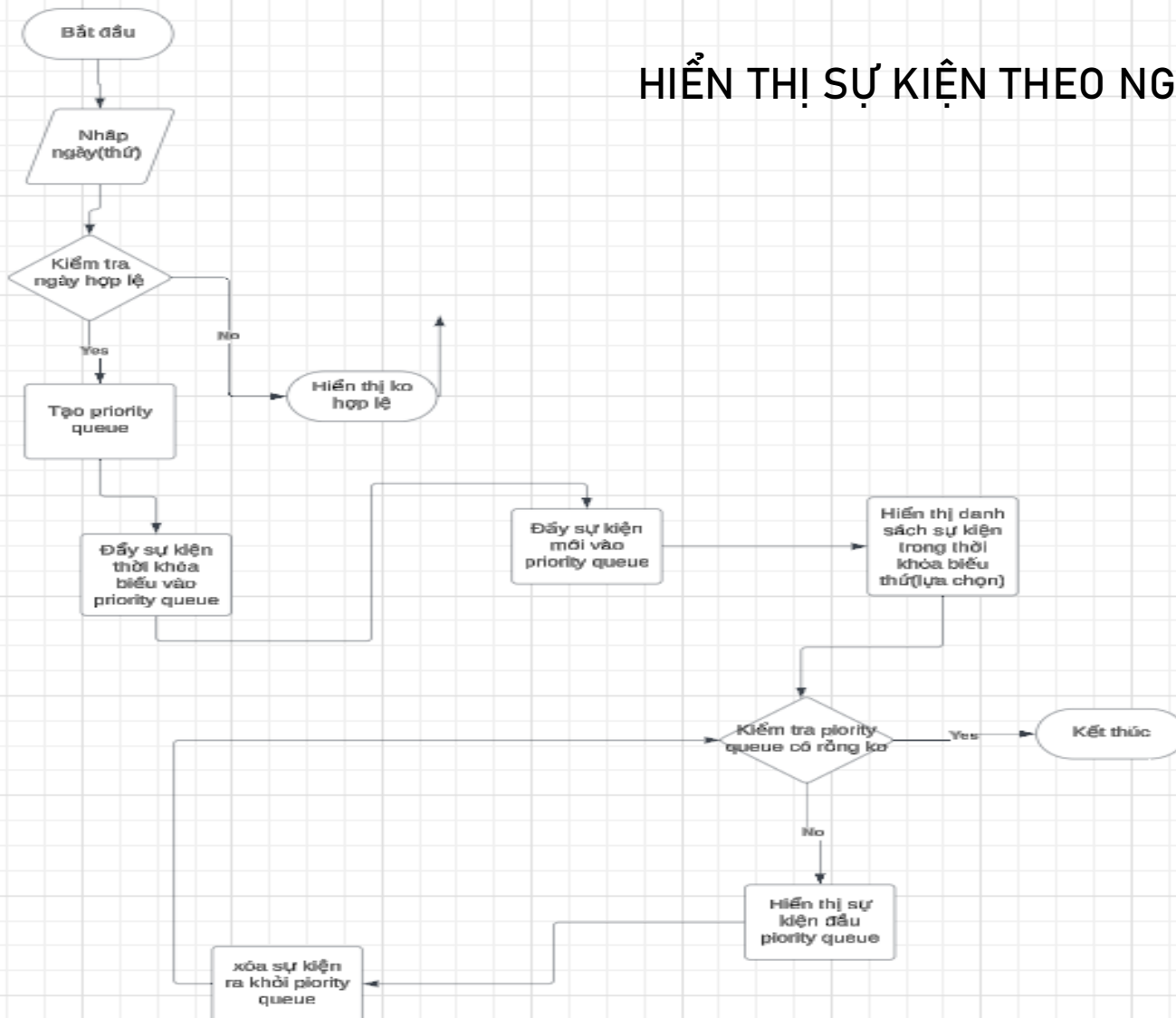
NHẬP THỜI KHOÁ BIỂU:



XỬ LÝ XUNG ĐỘT



HIỂN THỊ SỰ KIỆN THEO NGÀY



Môi trường lập trình và các thư viện

1. Môi trường lập trình: Visual studio code, VS code, GDB online,...

2. Thư viện sử dụng trong code

Thư viện chuẩn C++ (Standard Library - STL):

<iostream>: Hỗ trợ nhập/xuất dữ liệu từ console (cin, cout).

<string>: Hỗ trợ kiểu dữ liệu std::string cho việc xử lý chuỗi.

<vector>: Dùng cho cấu trúc dữ liệu danh sách động (std::vector).

<queue>: Dùng cho cấu trúc hàng đợi.

<algorithm>: Dùng các thuật toán như std::max, tính chiều cao cây.

<regex>: Xử lý chuỗi theo mẫu biểu thức chính quy (regex) trong hàm kiểm tra.

<limits>: Hỗ trợ lấy giá trị cực đại/cực tiểu của kiểu dữ liệu, sử dụng để xử lý lỗi nhập liệu. std::numeric_limits<int>::max().

```
#include <iostream>
#include <string>
#include <vector>
#include <queue>
#include <algorithm>
#include <regex>
#include <limits>
using namespace std;
```

4. Kết quả thực nghiệm

a. Cách xây dựng bộ test

Xác định các loại test case:

- **Test case dương tính:** Kiểm tra các trường hợp đầu vào hợp lệ và đảm bảo hệ thống trả về kết quả đúng.
- **Test case âm tính:** Kiểm tra các trường hợp đầu vào không hợp lệ và đảm bảo hệ thống xử lý lỗi một cách chính xác.
- **Test case giới hạn:** Kiểm tra các giá trị biên của đầu vào để đảm bảo hệ thống hoạt động ổn định.
- **Test case ngẫu nhiên:** Kiểm tra hệ thống với các dữ liệu ngẫu nhiên để phát hiện các lỗi khó dự đoán quan trọng trong quá trình phát triển phần mềm.

Xác định dữ liệu đầu vào:

- Dữ liệu hợp lệ: Phù hợp với yêu cầu của hệ thống.
- Dữ liệu không hợp lệ: Không phù hợp với yêu cầu của hệ thống
- Dữ liệu giới hạn: Các giá trị biên của phạm vi cho phép.

Xác định các bước thực hiện:

- Mô tả chi tiết các bước thực hiện để kiểm tra chức năng.

Xác định kết quả mong đợi:

- Xác định rõ kết quả mà hệ thống phải trả về trong từng trường hợp.

4. Kết quả thực nghiệm

b. Các kết quả test

Test case 1: Nhập thời khóa biểu hợp lệ

- Dữ liệu đầu vào:
 - Thứ 2:
 - Sự kiện 1: "bao cao" (09:00 - 10:00)
 - Sự kiện 2: "thuc hanh" (13:00 - 15:00)
 - ... (Nhập TKB cho các ngày còn lại)
- Kết quả mong đợi:
 - Lưu trữ thành công TKB theo từng ngày.

Test case 2: Nhập thời khóa biểu với giờ không hợp lệ

- Dữ liệu đầu vào:
 - Thứ 2:
 - Sự kiện 1: "thuc hanh" (15:00 - 25:00)
- Kết quả mong đợi:
 - Thông báo lỗi định dạng thời gian và yêu cầu nhập lại.

Test case 3: Thêm sự kiện mới hợp lệ

- Dữ liệu đầu vào:
 - Thứ 3: Thêm "thuc hanh" (11:00 - 12:00)
- Kết quả mong đợi:
 - Thêm sự kiện mới vào lịch thứ 4.

```
Nhap thoi gian cho thu 2
Nhap so su kien trong ngay: 2
Nhap mo ta su kien 1: bao cao
Nhap gio bat dau (dinh dang 24h, VD: 14:20): 09:00
Nhap gio ket thuc (dinh dang 24h, VD: 14:20): 10:00
Nhap mo ta su kien 2: thuc hanh
Nhap gio bat dau (dinh dang 24h, VD: 14:20): 13:00
Nhap gio ket thuc (dinh dang 24h, VD: 14:20): 15:00
Nhap thoi gian cho thu 3
Nhap so su kien trong ngay: 1
Nhap mo ta su kien 1: chay bo
Nhap gio bat dau (dinh dang 24h, VD: 14:20): 14:00
Nhap gio ket thuc (dinh dang 24h, VD: 14:20): 15:00
Nhap thoi gian cho thu 4
Nhap so su kien trong ngay: 0
Nhap thoi gian cho thu 5
Nhap so su kien trong ngay: 0
Nhap thoi gian cho thu 6
Nhap so su kien trong ngay: 0
Nhap thoi gian cho thu 7
Nhap so su kien trong ngay: 0
```

```
=== Quan Ly Su Kien ===
1. Them su kien
2. Hien thi su kien
3. Thoat
Nhap lua chon:
```

```
=== Quan Ly Su Kien ===
1. Them su kien
2. Hien thi su kien
3. Thoat
Nhap lua chon: 1
Nhap thu (1: Thu 2, 2: Thu 3, ..., 6: Thu 7): 2
Nhap mo ta su kien: thuc hanh
Nhap gio bat dau (dinh dang 24h, VD: 14:20): 11:00
Nhap gio ket thuc (dinh dang 24h, VD: 14:20): 12:00
Su kien moi (thuc hanh) trung voi su kien: hoc bai
Thoi gian su kien moi: 11:00 -> 12:00
Thoi gian su kien trung lap: 11:00 -> 12:00
Vui long nhap do uu tien cua su kien moi (1->10): 5
```

4. Kết quả thực nghiệm

b. Các kết quả test

Test case 4: Thêm sự kiện mới trùng lịch

Dữ liệu đầu vào:

- Thứ 3: Thêm sự kiện "di học" (14:00 - 15:00)

Kết quả mong đợi:

- Thông báo sự kiện bị trùng lịch.
- Nhập độ ưu tiên cho sự kiện mới (1-10).

Test case 5: Thêm sự kiện mới với độ ưu tiên

Dữ liệu đầu vào:

- Thứ 3: "chạy bo" (14:00 - 15:00)
- Thêm sự kiện "di học" (14:00 - 15:00) với độ ưu tiên 10.

Kết quả mong đợi:

- Thông báo sự kiện "di học" có độ ưu tiên cao hơn và được thêm vào lịch thứ 3, thay thế phần thời gian của sự kiện "chạy bo".

Test case 6: Hiển thị lịch theo thứ

Dữ liệu đầu vào:

- Chọn thứ 2 để hiển thị lịch.

Kết quả mong đợi:

- Chương trình in ra danh sách các sự kiện theo thứ tự thời gian bắt đầu

```
=== Quan Ly Su Kien ===
1. Them su kien
2. Hien thi su kien
3. Thoat
Nhap lua chon: 1
Nhap thu (1: Thu 2, 2: Thu 3, ..., 6: Thu 7): 2
Nhap mo ta su kien: di hoc
Nhap gio bat dau (dinh dang 24h, VD: 14:20): 14:00
Nhap gio ket thuc (dinh dang 24h, VD: 14:20): 15:00
Su kien moi (di hoc) trung voi su kien: chay bo
Thoi gian su kien moi: 14:00 -> 15:00
Thoi gian su kien trung lap: 14:00 -> 15:00
Vui long nhap do uu tien cua su kien trong thoi khoa bieu (1->10): 7
Vui long nhap do uu tien cua su kien moi (1->10): 10
Xet do uu tien...
Them su kien moi vao thoi khoa bieu.
```

```
=== Quan Ly Su Kien ===
1. Them su kien
2. Hien thi su kien
3. Thoat
Nhap lua chon: 2

=== Hien Thi Su Kien ===
Nhap thu (1: Thu 2, 2: Thu 3, ..., 6: Thu 7): 1
Danh sach su kien trong thoi khoa bieu thu 2:
bao cao - 09:00 -> 10:00 - Do uu tien: 10
tieng anh - 09:30 -> 11:00 - Do uu tien: 9
thuc hanh - 13:00 -> 15:00
```


5. Kết luận

Mức độ hoàn thành chung

- **Hoàn Thành Các Yêu Cầu Cơ Bản:** Nhóm đã đáp ứng đầy đủ các yêu cầu đặt ra cho dự án, bao gồm việc quản lý sự kiện, kiểm tra xung đột thời gian và hiển thị thời khóa biểu một cách trực quan và dễ hiểu.
- **Áp Dụng Hiệu Quả Các Giải Thuật và Cấu Trúc Dữ Liệu:** Cây AVL được sử dụng hiệu quả để duy trì cân bằng và tối ưu hóa tìm kiếm, thêm và xóa sự kiện, giúp hệ thống hoạt động nhanh chóng và ổn định ngay cả khi số lượng sự kiện lớn. Hàng đợi giúp quản lý và xử lý các sự kiện bị xung đột một cách logic và hiệu quả, đảm bảo không bỏ sót sự kiện nào.
- **Tính Toàn Vẹn và Tối Ưu Của Mã Nguồn:** Mã nguồn được viết rõ ràng, dễ hiểu và có tính tổ chức cao, giúp việc bảo trì và nâng cấp trong tương lai trở nên dễ dàng hơn. Nhóm đã chú trọng đến việc kiểm thử và phát hiện lỗi sớm, giúp cải thiện chất lượng sản phẩm cuối cùng.
- **Phối Hợp Nhóm Hiệu Quả:** Các thành viên trong nhóm đã phối hợp với nhau một cách ăn ý, mỗi người đều hoàn thành tốt nhiệm vụ được giao.
- **Khả Năng Ứng Dụng Thực Tế:** Hệ thống quản lý sự kiện và thời khóa biểu này có thể được áp dụng vào nhiều tình huống thực tế, từ việc quản lý lịch trình cá nhân cho đến tổ chức các sự kiện lớn, cho thấy tính linh hoạt và tiềm năng phát triển của dự án.

5. Kết luận

Những khó khăn gặp phải và cách giải quyết

Nhóm đã gặp nhiều khó khăn trong việc tìm được cấu trúc và thuật toán phù hợp để áp dụng vào bài toán sao cho có thể đáp ứng nhu cầu nghiên cứu sát với bài học.

Xây dựng bộ test có thể kiểm tra được tất cả những tình huống có thể gây ra lỗi hoặc kết quả không như mong đợi rất phức tạp và tốn thời gian, sau mỗi lần phát hiện lỗi, sẽ đều phải họp để đề ra phương án giải quyết

=> Nhờ khả năng làm việc nhóm cũng như phân chia công việc hợp lý nhóm đã hoàn thành nhiệm vụ được giao.

Hạn chế còn tồn tại và hướng phát triển.

Hạn chế: - Quản lý thời gian ngắn hạn: Chỉ quản lý được trong 1 tuần thay vì một tháng.

- Giao diện người dùng: Giao diện chưa tối ưu, gây khó khăn cho người dùng.

Giải pháp, cải tiến và ý tưởng phát triển

Cải thiện giao diện trực quan, dễ sử dụng.

Bổ sung tính năng nhập xuất excel.

Thêm các tính năng phụ: ví dụ: Với các app dành cho sinh viên như ICTSV có thể bổ sung tính năng tối ưu hóa hoạt động sao cho sinh viên có thể đạt nhiều điểm rèn luyện nhất (DP&Greedy) (DP: Tối ưu hóa số lượng sự kiện không bị trùng lặp với lịch học hoặc tối ưu hóa trọng số (vd lịch sự kiện quan trọng hơn lịch học))

A large, stylized graphic on the left side of the slide. It consists of a red background with a circular pattern of white dots of varying sizes, creating a sense of depth and movement. The word "HUST" is written in white, bold, sans-serif capital letters in the center of this graphic.

HUST

THANK YOU !