

**TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN TIN HỌC
ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG XẾP LỊCH THI
ĐẦU MÔN CỜ TƯỚNG

Người hướng dẫn: **TS. Trần Hồ Thủy Tiên**
Sinh viên thực hiện:

- 1. Nguyễn Thái Pháp - 123210067**
- 2. Lê Minh Mạnh - 123210063**
- 3. Lê Trần Duy Tân - 123210173**
- 4. Trần Văn Quyết – 123210171**

LỚP: 21PFIEV2
NHÓM: 21N88

Đà Nẵng, 05/2025.

DANH MỤC HÌNH VẼ	
MỞ ĐẦU	
1. TỔNG QUAN ĐỀ TÀI.....	1
2. CƠ SỞ LÝ THUYẾT.....	1
2.1. Ý tưởng	1
2.2. Cơ sở lý thuyết.....	1
3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN	1
3.1. Phát biểu bài toán.....	1
3.2. Cấu trúc dữ liệu	2
3.3. Thuật toán	3
4. CHƯƠNG TRÌNH VÀ KẾT QUẢ.....	8
4.1. Tổ chức chương trình	8
4.2. Ngôn ngữ cài đặt.....	9
4.3. Kết quả.....	9
4.3.1. Giao diện chính của chương trình.....	9
4.3.2. Kết quả thực thi của chương trình	10
4.3.3. Nhận xét đánh giá	12
5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	12
5.1. Kết luận.....	12
5.2. Hướng phát triển	12
TÀI LIỆU THAM KHẢO	12
PHỤ LỤC	13

MỞ ĐẦU

- **Mục đích thực hiện đề tài:**

Đề tài nhằm xây dựng một ứng dụng hỗ trợ **quản lý và xếp lịch thi đấu cờ tướng một cách tự động**, từ đó giúp tối ưu hóa quá trình tổ chức giải đấu, theo dõi kết quả thi đấu và xác định nhà vô địch một cách minh bạch, nhanh chóng và hiệu quả.

- **Mục tiêu đề tài:**

- Xây dựng hệ thống **xếp cặp thi đấu công bằng** dựa trên kết quả của các trận đấu trước đó.
- Quản lý hiệu quả **danh sách vận động viên (VĐV)**, **kết quả các trận đấu** và **bảng xếp hạng** hiện tại.
- Thực hiện **loại VĐV tự động** theo **thể thức thi đấu loại trực tiếp**.

- **Phạm vi và đối tượng nghiên cứu:**

- Ứng dụng được thiết kế để phục vụ cho các **giải đấu cờ tướng** có: **Số lượng VĐV là số chẵn**.
- **Số trận đấu tối đa là k**, với k do người dùng nhập vào khi bắt đầu giải.

- **Phương pháp nghiên cứu:**

- Sử dụng thuật toán **xếp cặp đấu dựa vào điểm số theo thể thức Thụy Sĩ** nhằm đảm bảo tính công bằng và hợp lý trong việc phân cặp thi đấu.
- Kết hợp các **cấu trúc dữ liệu phù hợp** để lưu trữ và xử lý thông tin.
- Áp dụng kỹ thuật **xử lý file** để đảm bảo việc lưu trữ và truy xuất dữ liệu linh hoạt và hiệu quả.

- **Cấu trúc của đồ án môn học:**

1. TỔNG QUAN ĐỀ TÀI

- Mục đích: Xây dựng ứng dụng quản lý và xếp lịch thi đấu cờ tướng tự động, hỗ trợ tối ưu hóa quá trình tổ chức giải đấu.
- Phạm vi: Áp dụng cho các giải đấu có số lượng VĐV chẵn, xếp cặp dựa trên kết quả các trận trước.
- Đối tượng: Ban tổ chức giải đấu và các VĐV tham gia.
- Phương pháp: Sử dụng thuật toán xếp cặp theo điểm số, chuỗi thắng/thua và cơ chế dịch chuyển bàn thi đấu.

2. CƠ SỞ LÝ THUYẾT

2.1. Ý tưởng

- Xếp cặp ngẫu nhiên cho trận đầu tiên.
- Từ trận thứ 2, xếp cặp dựa trên kết quả trận trước: Thắng gặp Thắng, Thua gặp Thua, nếu số VĐV thi đấu bị lẻ khi xếp cặp theo kết quả đấu, chương trình sẽ tìm VĐV có kết quả thi đấu gần nhất để xếp cặp.
- Loại VĐV dựa trên số trận thua liên tiếp và tổng điểm (2 VĐV có điểm số bằng nhau thì ưu tiên VĐV vừa thắng trận gần nhất sẽ được giữ lại).
- Đảm bảo loại dần các VĐV có kết quả thi đấu thấp khi đến trận cuối chỉ còn lại 2 bàn đấu để đấu trận quyết định người vô địch.

2.2. Cơ sở lý thuyết

- Xử lý chuỗi: Đọc/ghi danh sách VĐV từ file .txt hoặc nhập từ bàn phím, kiểm tra tính hợp lệ của đầu vào.
- Cấu trúc dữ liệu: Sử dụng struct Player để lưu thông tin VĐV (tên, điểm, lịch sử trận đấu).
- Thuật toán:
 - o Xếp cặp ngẫu nhiên bằng random_shuffle.
 - o Sắp xếp VĐV theo điểm số và số trận thắng, thua.
 - o Cơ chế loại VĐV có nhiều trận thua sau vòng đấu thứ 3.

3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

3.1. Phát biểu bài toán

Mô tả đầu vào (Input) và đầu ra (Output)

- **Đầu vào:**
 - Danh sách VĐV (nhập từ bàn phím hoặc file .txt).
 - Số trận tối đa k.
 - Kết quả từng trận (thắng/thua).
- **Đầu ra:**
 - Danh sách cặp đấu theo từng bàn.
 - Bảng xếp hạng sau mỗi trận và VĐV vô địch.

3.2. Cấu trúc dữ liệu

a) Struct (Player)

- **Mô tả:** Được sử dụng để lưu trữ thông tin của một vận động viên (VĐV) tham gia giải đấu.
- **Các trường:**
 - name: Tên VĐV (kiểu string).
 - daThang: Kết quả trận gần nhất (kiểu bool).
 - daBiLoai: Trạng thái bị loại (kiểu int, thực chất hoạt động như bool).
 - soTranThang, soTranThua, diem: Số trận thắng, thua và điểm số (kiểu int).
 - xepHang, vongBiLoai: Hạng và vòng bị loại (kiểu int).
 - lichSuThang: Lưu lịch sử thắng/thua của VĐV (kiểu vector<bool>).
 - doiThuDaGap: Lưu danh sách đối thủ đã gặp (kiểu vector<string>).
- **Vai trò:** Là cấu trúc trung tâm để quản lý thông tin và trạng thái của các VĐV.

b) Vector

- **Sử dụng:**
 - vector<Player> danhSachVDV: Lưu danh sách tất cả VĐV.
 - vector<Player> danhSachBiLoai: Lưu danh sách VĐV đã bị loại.
 - vector<vector<pair<int, int>>> lichSuTranDau: Lưu lịch sử các cặp đấu theo từng trận (mỗi trận là một vector<pair<int, int>> chứa các cặp chỉ số VĐV).

- `vector<vector<pair<int, int>>>` `cacCapDauHienTai`: Lưu các cặp đấu hiện tại.
- `vector<int>` `danhSachConLai`: Lưu chỉ số của các VĐV chưa bị loại để xếp cặp.
- `vector<bool>` `lichSuThang` (trong `Player`): Lưu lịch sử thắng/thua của từng VĐV.
- `vector<string>` `doiThuDaGap` (trong `Player`): Lưu danh sách tên đối thủ đã gặp.
- **Vai trò:** Là cấu trúc dữ liệu chính để quản lý danh sách động, hỗ trợ thêm, xóa, truy cập ngẫu nhiên và duyệt dễ.

c) Pair

- **Sử dụng:** `pair<int, int>` được dùng để biểu diễn cặp đấu giữa hai VĐV (lưu hai chỉ số của VĐV trong `danhSachVDV`).
- **Vai trò:** Đơn giản hóa việc lưu trữ và xử lý các cặp đấu trong `lichSuTranDau` và `cacCapDauHienTai`.

d) Map

- **Sử dụng:** `map<int, vector<int>>` `nhomTheoTrangThai` trong hàm `XepCapTranDauNgauNhiem` để nhóm VĐV theo số trận thắng.
- **Vai trò:** Hỗ trợ tổ chức VĐV thành các nhóm dựa trên số trận thắng, giúp xếp cặp đấu một cách hợp lý.

e) String

- **Sử dụng:** Lưu trữ tên VĐV, tên file, và các chuỗi hiển thị giao diện (như tiêu đề, hướng dẫn).
- **Vai trò:** Quản lý dữ liệu dạng văn bản.

f) Array tĩnh

- **Sử dụng:** Mảng string để lưu các tùy chọn menu (`options[]`) và các chuỗi ASCII art (như `championsArt[]`).
- **Vai trò:** Hỗ trợ hiển thị giao diện người dùng.

3.3. Thuật toán

Trình bày các thuật toán và phân tích độ phức tạp của các thuật toán.

a) Sắp xếp (Sorting)

- **Thuật toán:** Sử dụng hàm sort từ thư viện <algorithm> với các tiêu chí tùy chỉnh (sử dụng lambda function).
- **Ứng dụng:**
 - o Trong XepCapTranDauNgauNhiem:
 - Sắp xếp VĐV theo số trận thắng giảm dần, số trận thua tăng dần để xếp cặp.
 - Sắp xếp các cặp đấu theo số trận thắng tối thiểu của cặp.
 - o Trong HienThiBangXepHang:
 - Sắp xếp VĐV theo số trận thắng giảm dần và chuỗi thua tăng dần.
 - o Trong LoaiVanDongVien:
 - Sắp xếp VĐV theo số trận thua giảm dần và chuỗi thua giảm dần để chọn VĐV bị loại.
- **Độ phức tạp:** $O(n \log n)$, với n là số phần tử cần sắp xếp.

Ký hiệu:

- o $n = 20$: Số vận động viên (VĐV) ban đầu.
- o $soTranToiDa = 7$: Số trận tối đa.
- o m : Số VĐV chưa bị loại (giảm dần, ví dụ: $20 \rightarrow 4$).
- o k : Số cặp đấu ($\approx m/2$, ví dụ: $10 \rightarrow 2$).

Ước lượng số phép toán

Hàm:

- XepCapTranDauNgauNhiem: Sắp xếp danhSachConLai (m) và cacCapDau (k).
- HienThiBangXepHang: Sắp xếp sapXep ($n = 20$).
- LoaiVanDongVien: Sắp xếp danhSachVDV ($n = 20$).

Tính toán:

- Công thức: $2 * n * \log_2(n)$ (so sánh và hoán đổi).
- XepCapTranDauNgauNhiem (7 vòng):
 - o Vòng 1: $m = 20, k = 10$
 $\rightarrow 2 * 20 * \log_2(20) \approx 172, 2 * 10 * \log_2(10) \approx 66 \rightarrow 238.$
 - o Vòng 7: $m = 4, k = 2$
 $\rightarrow 2 * 4 * \log_2(4) = 16, 2 * 2 * \log_2(2) = 4 \rightarrow 20.$
 - o Trung bình: ~ 130 phép toán/vòng $\rightarrow 7 * 130 = 910.$

- HienThiBangXepHang (gọi 5 lần):
 $2 * 20 * \log_2(20) \approx 172 \rightarrow 5 * 172 = 860$.
- LoaiVanDongVien (gọi 3 lần, vòng 3-5): $3 * 172 = 516$.
- Tổng: $910 + 860 + 516 = 2,286$.

b) Xáo trộn ngẫu nhiên (Random Shuffle)

- **Thuật toán:** Sử dụng `random_shuffle` từ <algorithm> để xáo trộn ngẫu nhiên các VĐV trong một nhóm.
- **Ứng dụng:** Trong `XepCapTranDauNgauNhiem` để đảm bảo tính ngẫu nhiên khi xếp cặp VĐV cùng số trận thắng.
- **Độ phức tạp:** $O(n)$, với n là số phần tử.

Hàm: `XepCapTranDauNgauNhiem`.

Tính toán:

- Công thức: $2 * m$ (so sánh và hoán đổi).
- Vòng 1: $m = 20 \rightarrow 2 * 20 = 40$.
- Vòng 7: $m = 4 \rightarrow 2 * 4 = 8$.
- Trung bình: ~ 24 phép toán/vòng $\rightarrow 7 * 24 = 168$.

c) Loại bỏ phần tử (Remove-erase idiom)

- **Thuật toán:** Sử dụng `remove_if` kết hợp với `erase` để loại bỏ VĐV đã bị đánh dấu `daBiLoai` khỏi `danhSachVDV` hoặc `danhSachConLai`.
- **Ứng dụng:**
 - Trong `XepCapTranDauNgauNhiem` và `LoaiVanDongVien` để cập nhật danh sách sau khi loại VĐV.
- **Độ phức tạp:** $O(n)$, với n là số phần tử.

Hàm:

- `XepCapTranDauNgauNhiem`: Loại VĐV khỏi `danhSachConLai`.
- `LoaiVanDongVien`: Loại VĐV khỏi `danhSachVDV`.

Tính toán:

- `remove_if` + `erase`: $m + \text{soVDVBiLoai}$.
- `XepCapTranDauNgauNhiem` (2 lần, vòng 3-4):
 - Vòng 3: $m = 16$, loại 4 $\rightarrow 16 + 4 = 20$.

- Vòng 4: $m = 12$, loại 4 $\rightarrow 12 + 4 = 16$.
- Tổng: $20 + 16 = 36$.
- LoạiVanDongVien (3 lần): $20 + 4 = 24$ mỗi lần $\rightarrow 3 * 24 = 72$.
- Tổng: $36 + 72 = 108$.

d) Duyệt tuyến tính (Linear Traversal)

- **Mô tả:** Duyệt qua danh sách VĐV, cặp đấu, hoặc lịch sử trận đấu để hiển thị, cập nhật, hoặc xử lý dữ liệu.
- **Ứng dụng:**
 - Hiển thị danh sách VĐV (HienThiDanhSachVDV).
 - Hiển thị cặp đấu (HienThiCacCapDau).
 - Cập nhật kết quả trận đấu (NhapKetQuaTranDau).
 - Tính chuỗi thua liên tiếp (tinhChuoiThua).
- **Độ phức tạp:** $O(n)$, với n là số phần tử.

Hàm:

- HienThiDanhSachVDV, HienThiCacCapDau, NhapKetQuaTranDau, tinhChuoiThua.

Tính toán:

- Mỗi phần tử: **~2 phép toán.**
- HienThiDanhSachVDV (5 lần): $2 * 20 = 40 \rightarrow 5 * 40 = 200$.
- HienThiCacCapDau (7 vòng, trung bình $k = 7$): $2 * 7 = 14 \rightarrow 7 * 14 = 98$.
- NhapKetQuaTranDau (7 vòng, trung bình $k = 7$): $2 * 7 = 14 \rightarrow 7 * 14 = 98$.
- tinhChuoiThua (gọi trong HienThiBangXepHang 5 lần, LoạiVanDongVien 3 lần):
 - Mỗi VĐV: $2 * 7 = 14 \rightarrow 14 * 20 = 280$ mỗi lần.
 - Tổng: $(5 + 3) * 280 = 2,240$.
- Tổng: $200 + 98 + 98 + 2,240 = 2,636$.

e) Quản lý giao diện Console

- **Mô tả:** Sử dụng các hàm như ToaDo, DoiMau, VeKhung, XoaVung để vẽ giao diện đồ họa trên console Windows.
- **Ứng dụng:**
 - o Vẽ khung, tiêu đề, cúp vàng (VeCupVang), và từ "Champions" (VeChuChampions).
 - o Hiện thị menu động (VeMenuChinh) và xử lý tương tác người dùng.
- **Độ phức tạp:** $O(1)$ cho mỗi thao tác vẽ (tùy thuộc vào kích thước vùng vẽ).

Hàm: VeKhung, XoaVung, VeCupVang, VeChuChampions.

Tính toán:

- VeKhung (giả sử $w = 50$, $h = 16$, gọi 20 lần): $2 * (50 + 16) = 132 \rightarrow 20 * 132 = 2,640$.
- XoaVung (20 lần): $50 * 16 = 800 \rightarrow 20 * 800 = 16,000$.
- VeCupVang, VeChuChampions (5 lần): $20 * 5 = 100$.
- **Tổng:** $2,640 + 16,000 + 100 = 18,740$.

f) Quản lý đầu vào/đầu ra file

- **Mô tả:** Đọc danh sách VĐV từ file văn bản sử dụng fopen, fgets, và xử lý chuỗi.
- **Ứng dụng:** Trong NhapDanhSachNguoiChoi để đọc danh sách VĐV từ file .txt.
- **Độ phức tạp:** $O(n)$, với n là số dòng trong file.

Hàm: NhapDanhSachNguoiChoi.

Tính toán: $10 * 20 = 200$ (đọc 20 dòng).

g) Quản lý trạng thái trò chơi

- **Mô tả:** Sử dụng các biến trạng thái (daNhapDanhSach, daXepCapTiepTheo, daQuaVong3, daXacDinhVoDich, daLoaiVDVTaiVongHienTai) để kiểm soát luồng chương trình.
- **Ứng dụng:** Quyết định hiển thị thông báo lỗi, chuyển hướng menu, hoặc thực hiện hành động phù hợp (như loại VĐV, xếp cặp, hoặc kết thúc giải đấu).
- **Độ phức tạp:** $O(1)$ cho mỗi kiểm tra trạng thái.

Tính toán: Giả sử 50 lần kiểm tra \rightarrow 50 phép toán.

Tổng số phép toán

Loại thao tác	Số phép toán
Sắp xếp	2,268
Xáo trộn	168
Loại bỏ phần tử	108
Duyệt tuyến tính	2,636
Giao diện Console	18,740
Đọc file	200
Kiểm tra trạng thái	50
Tổng	24,188

Không gian

- danhSachVDV, danhSachBiLoai, sapXep: **O(20)**.
- danhSachConLai, cacCapDauHienTai: **O(20)**.
- Chuỗi tạm (XoaVung): **O(50)**.
- **Tổng: O(20)**.

Kết luận

- **Thời gian:** $\sim 24,188$ phép toán, chi phối bởi giao diện console ($\sim 77\%$) và tính chuỗi thua ($\sim 9\%$).
- **Không gian:** **O(20)**.
- **Tối ưu hóa:** Giảm vẽ giao diện lặp lại, lưu kết quả chuỗi thua để tái sử dụng.

➔ Đặc điểm nổi bật:

- Tích hợp giao diện console đồ họa với các hiệu ứng màu sắc, khung, và ASCII art.
- Quản lý giải đấu phức tạp với các vòng đấu, loại VĐV, và xếp hạng.
- Xử lý lỗi đầu vào (như số VĐV không hợp lệ, file không tồn tại) và tương tác người dùng trực quan.

4. CHƯƠNG TRÌNH VÀ KẾT QUẢ

4.1. Tổ chức chương trình

- Ngôn ngữ: C++.
- Thư viện: `<iostream>`, `<vector>`, `<algorithm>`, `<windows.h>`.
- Chức năng chính:
 - Nhập danh sách VĐV và số trận thi đấu.

- Xếp cặp và nhập kết quả trận.
- Hiển thị bảng xếp hạng, danh sách VĐV bị loại.
- Thoát khỏi giải đấu

4.2. Ngôn ngữ cài đặt

Sử dụng ngôn ngữ C/C++

4.3. Kết quả

4.3.1. Giao diện chính của chương trình



4.3.2. Kết quả thực thi của chương trình

Mô tả kết quả thực hiện chương trình.

- Danh sách VĐV được nhập từ file hoặc bàn phím.

```
┌ DANH SACH VAN DONG VIEN ┐  
  
1. Nguyen Duc Binh  
2. Ha Van Khanh Dat  
3. Le Trung Phong  
4. Tran Van Dang  
5. Nguyen Huy Khang  
6. Le Van Dung  
7. Nguyen Cao Cuong  
8. Tran Quoc Anh  
9. Tran Van Quyet  
10. Le Minh Manh  
  
HUONG DAN:  
[R] Reset danh sach  
[1] Quay ve menu chinh  
[2] Xep cap tran dau
```

- Danh sách cặp đấu được hiển thị theo bàn.

```
| DANH SACH THI DAU TRAN 6/7 |  
  
Ban 1:Huỳnh Phúc Quang Minh vs Nguyễn Thị Hoàng Giang  
  
Ban 2:Khấu Thị Xuân Vũ vs Nguyễn Thị Hoàng Giang  
  
Ban 3:Đặng Thị Kim Liên vs Trương Hoàng Mỹ Ái  
  
Ban 4:Nguyễn Duy Khoa vs Lê Minh Mạnh  
  
Ban 5:Trần Dương Xuân Anh vs Nguyễn Trung Nghĩa  
  
NHAN:  
[1] Quay lai menu chinh  
[3] Nhap ket qua cac tran nay
```

STT	HỌ VÀ TÊN	TRẦN THĂNG	TRẦN THÚA	CHUỐI THUA
1	Huỳnh Phúc Quang Minh	5	1	0
2	Nguyễn Thị Hoàng Giang	5	2	0
3	Khấu Thị Xuân Vũ	5	1	1
4	Nguyễn Trung Nghĩa	4	2	0
5	Đặng Thị Kim Liên	4	2	0
6	Nguyễn Duy Khoa	4	2	0
7	Trần Dương Xuân Anh	4	2	1
8	Trương Hoàng Mỹ Ái	3	3	1
9	Lê Minh Mạnh	3	3	2

NHAN:

[1] Quay lại menu chính

[2] Xếp cấp trận tiếp theo

[5] Loại VDV sau vòng 3 (nếu có)

DANH SACH VDV BI LOAI				
STT	HO VA TEN	TRAN THANG	TRAN THUA	VONG BI LOAI
1	Le Minh Manh	0	3	3
2	Tran Van Quyet	1	2	3

NHAN:

NHAN:uay lai menu chinh

[1] Quay lai menu chinhhep theo

[2] Xep cap cho tran tiep theo

[illegible]

4.3.3. Nhận xét đánh giá

- Ưu điểm: Tự động hóa quy trình xếp lịch, xử lý được số lượng VDV lớn.
- Hạn chế: Chưa hỗ trợ giao diện đồ họa, chưa xử lý trường hợp VDV lẻ.
- Những cải tiến so với code được

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

- Ứng dụng đáp ứng yêu cầu cơ bản của đề tài, xử lý hiệu quả việc xếp lịch và quản lý giải đấu.

5.2. Hướng phát triển

- *Thêm tính năng xem chi tiết lịch sử trận đấu.*
- *Cải thiện giao diện người dùng (GUI).*
- *Xử lý trường hợp số VDV lẻ bằng cơ chế "bye".*

TÀI LIỆU THAM KHẢO

1. Tài liệu hướng dẫn C++.
2. Giáo trình Cấu trúc dữ liệu và Giải thuật.

PHỤ LỤC

Sinh viên bỏ Code từng phần vào đây. {Font: Time New Roman; thường; cỡ chữ: 12; dẫn dòng: 1,3; căn lề: justified}

- Mã nguồn đầy đủ: Xem file DoAnTinHoc.cpp.