



# APC ROAD TO OLP

*Tài liệu ôn tập Competitive Programming dành cho  
thành viên Câu lạc bộ Lập trình ứng dụng*

Câu lạc bộ Lập trình ứng dụng - Applied Programming Club

Câu lạc bộ trực thuộc UMT - Khoa Công nghệ

# Mục lục

Lời nói đầu	2
<b>1 TỔNG QUAN VỀ LẬP TRÌNH THI ĐẦU</b>	<b>3</b>
1.1 Giới thiệu về Lập trình thi đầu . . . . .	3
1.1.1 Lập trình thi đầu là gì? . . . . .	3
1.1.2 Lợi ích khi tham gia CP . . . . .	3
1.1.3 Các cuộc thi nổi bật trong CP . . . . .	4
1.1.4 Nền tảng luyện tập & OJ phổ biến . . . . .	4
1.2 Ngôn ngữ lập trình trong CP . . . . .	5
1.2.1 Vì sao C++ là lựa chọn hàng đầu? . . . . .	5
1.2.2 So sánh nhanh với Python, Java . . . . .	5
1.3 Cài đặt môi trường . . . . .	5
1.3.1 Cài đặt g++/MinGW, VS Code . . . . .	5
1.3.2 Thiết lập template C++ cho CP . . . . .	5
1.4 Giới thiệu các nền tảng luyện tập . . . . .	5
<b>2 GIỚI THIỆU VỀ LẬP TRÌNH</b>	<b>6</b>
<b>3 GIỚI THIỆU VỀ LẬP TRÌNH</b>	<b>7</b>
<b>4 GIỚI THIỆU VỀ LẬP TRÌNH</b>	<b>8</b>
<b>5 GIỚI THIỆU VỀ LẬP TRÌNH</b>	<b>9</b>
<b>6 GIỚI THIỆU VỀ LẬP TRÌNH</b>	<b>10</b>
<b>7 GIỚI THIỆU VỀ LẬP TRÌNH</b>	<b>11</b>
<b>8 GIỚI THIỆU VỀ LẬP TRÌNH</b>	<b>12</b>
<b>9 GIỚI THIỆU VỀ LẬP TRÌNH</b>	<b>13</b>
<b>10 GIỚI THIỆU VỀ LẬP TRÌNH</b>	<b>14</b>

# LỜI NÓI ĐẦU

Trong thời đại công nghệ số, lập trình không chỉ là một kỹ năng quan trọng mà còn là cánh cửa mở ra vô vàn cơ hội phát triển cho sinh viên trong lĩnh vực công nghệ thông tin và khoa học máy tính. Đặc biệt, với các cuộc thi lập trình như **Olympic Tin học Sinh viên Việt Nam (OLP)**, **International Collegiate Programming Contest (ICPC)** hay các kỳ thi trực tuyến toàn cầu, lập trình thi đấu (*Competitive Programming - CP*) đã trở thành môi trường tuyệt vời để sinh viên rèn luyện tư duy logic, kỹ năng phân tích và khả năng giải quyết vấn đề.

Với khát vọng tạo dựng một cộng đồng học thuật năng động tại Trường Đại học Quản lý và Công nghệ TP.HCM (UMT) và hỗ trợ sinh viên UMT tiếp cận *Competitive Programming* từ con số 0, **Câu lạc bộ Lập trình Ứng dụng (APC)** trực thuộc UMT - Khoa Công nghệ đã biên soạn bộ tài liệu này với mong muốn đồng hành cùng sinh viên từ những bước đi đầu tiên trong hành trình lập trình thi đấu.

**Tài liệu được thiết kế với ba định hướng chính:**

- **Dễ tiếp cận cho người mới bắt đầu:** Nội dung bắt đầu từ những kiến thức lập trình cơ bản nhất, không yêu cầu nền tảng trước, giúp sinh viên học từ con số 0.
- **Hệ thống kiến thức nền tảng và nâng cao:** Bao quát các chủ đề từ **Lập trình cơ bản**, **Cấu trúc dữ liệu** đến **Thuật toán**, tạo nền móng vững chắc cho việc tham gia lập trình thi đấu và phát triển kỹ năng lập trình chuyên sâu.
- **Hướng đến thực hành và thi đấu:** Cung cấp bài tập và phân tích chiến lược giải bài, giúp sinh viên không chỉ học lý thuyết mà còn rèn luyện khả năng tư duy và phản xạ trong môi trường thi đấu thực tế.

Tài liệu này không chỉ là nguồn học tập, mà còn là cầu nối gắn kết các thành viên trong CLB APC, tạo ra một cộng đồng học thuật năng động, nơi các bạn sinh viên cùng nhau học hỏi, trao đổi và tiến bộ. Nhóm biên soạn hy vọng rằng với sự đồng hành của tài liệu này, mỗi bạn sinh viên đều có thể từng bước vươn lên, từ người mới bắt đầu đến thí sinh tự tin tham gia các kỳ thi lập trình, góp phần nâng cao vị thế của UMT trên các sân chơi học thuật trong và ngoài nước.

Trong quá trình biên soạn, nhóm đã nỗ lực hết mình để đảm bảo tính chính xác và tính thực tiễn. Tuy nhiên, do phạm vi kiến thức rộng lớn, khó tránh khỏi những thiếu sót. Chúng tôi rất mong nhận được những ý kiến đóng góp từ bạn đọc để tài liệu ngày càng hoàn thiện hơn.

TP. Hồ Chí Minh, Ngày 6 tháng 11 năm 2025

**Nhóm biên soạn tài liệu APC**

# CHƯƠNG 1

## TỔNG QUAN VỀ LẬP TRÌNH THI ĐẤU

Người viết: Đặng Phúc An Khang

Tài liệu tham khảo: Nhập môn Lập trình - Khoa CNTT, HCMUS

### 1.1 Giới thiệu về Lập trình thi đấu

#### 1.1.1 Lập trình thi đấu là gì?

**Lập trình thi đấu (Competitive Programming - CP)** là hình thức giải các bài toán thuật toán (làm bài trên máy tính) trong một khoảng thời gian hữu hạn (thường từ 1,5 đến 5 giờ), dưới những *ràng buộc chặt chẽ* về thời gian chạy và bộ nhớ. Thí sinh (cá nhân hoặc đội) viết chương trình:

- **Đọc** dữ liệu từ bàn phím/file và **ghi** kết quả ra màn hình/file.
- **Được chấm tự động** trên bộ test ẩn bởi hệ thống *Online Judge* (OJ) hoặc chấm offline với các kỳ thi như Học sinh giỏi Quốc gia (HSGQG).
- **Được đánh giá** theo tiêu chí chính: *đúng* (correctness), *nhANH* (time complexity/performance), *tiết kiệm bộ nhớ* (space usage), đôi khi có *điểm từng phần* (partial score).

**Một số thuật ngữ & verdict thường gặp:**

- **AC** (Accepted): Kết quả đúng trên tất cả test.
- **WA** (Wrong Answer): Sai kết quả trên ít nhất một test.
- **TLE** (Time Limit Exceeded): Vượt quá giới hạn thời gian.
- **MLE** (Memory Limit Exceeded): Vượt quá giới hạn bộ nhớ.
- **RE** (Runtime Error): Lỗi khi chạy (chia cho 0, truy cập ngoài mảng, ...).
- **CE** (Compilation Error): Lỗi biên dịch.

#### 1.1.2 Lợi ích khi tham gia CP

Lập trình thi đấu không chỉ là “giải cho đúng rồi lấy điểm”. Đây là môi trường giúp người học rèn luyện rất nhiều kỹ năng quan trọng theo cách rõ ràng và thực tế. Trước hết, bạn học cách *nghĩ có trình tự*: tách một bài toán thành các bước nhỏ, chọn cách làm hợp lý (thuật toán) và chọn cách lưu trữ dữ liệu phù hợp (cấu trúc dữ liệu). Bạn cũng hình thành thói quen ước lượng thời gian chạy để chương trình không bị **TLE** (chạy quá thời gian) và viết mã gọn gàng, dễ đọc để dễ *debug* (tìm và sửa lỗi), hạn chế lỗi thường gặp như lệch chỉ số, tràn số hay sai số khi tính toán. Trong lúc thi, bạn học cách quản lý thời gian: làm bài dễ trước để “mở khoá” điểm, phân bổ thời lượng hợp lý và giữ bình tĩnh khi gặp **WA** (sai đáp án) hoặc các lỗi khác. Nếu thi theo đội (như ICPC, đội 3 người dùng chung 1 máy), bạn còn rèn kỹ năng làm việc nhóm: phân vai rõ ràng (người đọc đề, người nghĩ thuật toán, người code), thống nhất *template* (khung mã dùng sẵn) và tự kiểm thử trước khi nộp. Nhờ đó, bạn xây nền tảng **DSA** (Cấu trúc dữ liệu & Giải thuật) vững chắc, có lợi thế khi phỏng vấn thực tập/việc làm, đồng thời có cơ hội tham gia các sân chơi, học bổng và giao lưu quốc tế. Cuối cùng, CP tạo cho bạn kỷ luật tự học và tư duy phản biện: luôn kiểm chứng lời giải, tự tạo ví dụ phản bác để “bắt lỗi” chương trình, từ đó dần dần tìm ra cách giải đúng và tốt hơn.

### 1.1.3 Các cuộc thi nổi bật trong CP

#### 1.1.3.1 Quốc tế

- **ICPC (International Collegiate Programming Contest)** — Cuộc thi đồng đội gồm **3 sinh viên/đội** dùng **1 máy tính** trong **4–5 giờ**, giải khoảng **8–13 bài**. Xếp hạng dựa trên **số bài giải đúng (AC)**, hoà giải bằng **penalty time**: thời gian đến lần nộp chấp nhận đầu tiên của mỗi bài cộng **20 phút phạt cho mỗi lần nộp sai** trước đó; bài không giải được thì *không tính thời gian*.<sup>1</sup> Điều kiện dự thi nêu rõ giới hạn số lần tham dự vòng khu vực/thế giới theo từng mùa và điều kiện sinh viên.<sup>2</sup>
- **Codeforces Rounds/Global Rounds** — Hệ thống thi **cá nhân** trực tuyến; thời lượng thường **1.5–2 giờ**, nhiều bài từ **A (dễ) đến F/G (khó)**. Kết quả các vòng **ảnh hưởng rating** cá nhân; cơ chế tính điểm/xếp hạng và một số chi tiết chấm điểm (đúng/sai, đôi khi có điểm động theo thời gian ở các format nhất định) được công bố trên blog quy tắc/chấm điểm của Codeforces. Đây là môi trường luyện tốc độ đọc đề, lựa chọn chiến lược làm bài và giữ phong độ ổn định qua các vòng xếp hạng định kỳ.

#### 1.1.3.2 Trong nước

- **OLP Tin học Sinh viên Việt Nam (OLP)** — Sự kiện học thuật quốc gia tổ chức **định kỳ cuối năm (tháng 12)** bên cạnh các hoạt động ICPC Việt Nam. OLP có nhiều khối thi (Chuyên, Không chuyên, Siêu cúp, Nguồn mở...), thể thức chấm có thể **đúng/sai hoặc theo subtask** tùy nội dung, và là sân chơi phù hợp cho sinh viên xây nền DSA và cọ xát học thuật trong nước.
- **ICPC Việt Nam** — Hệ sinh thái **đội 3 người** theo chuẩn ICPC quốc tế: **vòng sơ loại trực tuyến** và **vòng khu vực/quốc gia onsite**. Xếp hạng theo **số bài AC + penalty time**, là cửa ngõ để các đội tiến tới các vòng khu vực châu Á và xa hơn. Lịch thi và điều lệ cập nhật hàng năm theo thông báo chính thức.
- **VNOI CUP** — Giải lập trình thường niên do cộng đồng VNOI tổ chức. Vòng loại trực tuyến trên **VNOJ** thường có **180 phút** với khoảng **7 bài**; **chung kết** diễn ra **onsite**, chọn các thí sinh điểm cao nhất từ vòng loại. Thể thức từng năm có thể điều chỉnh (số bài, cách chấm điểm/điểm phần), lịch và điều lệ được công bố trước mùa giải.

*Lưu ý: Thể lệ, lịch thi và thể thức có thể thay đổi theo từng mùa giải/đơn vị tổ chức; hãy luôn kiểm tra trang chính thức để cập nhật mới nhất.*

### 1.1.4 Nền tảng luyện tập & OJ phổ biến

#### 1.1.4.1 Quốc tế

- **CSES Problem Set** — Bộ sưu tập bài tập có cấu trúc lộ trình (Introductory, Sorting & Searching, DP, Graph, Geometry, ...) nhằm **xây nền thuật toán** theo tiến trình, thích hợp cho người mới đến trung cấp; không có rating/thi định kỳ, tập trung vào chất lượng đề và tính hệ thống.
- **SPOJ (Sphere Online Judge)** — Hệ thống *online judge* lâu năm với ~20,000+ bài, hỗ trợ ≥40 ngôn ngữ, cho phép tổ chức contest riêng và nộp bài bất kỳ lúc nào. Phù hợp để **đa dạng hoá luyện tập** theo chủ đề, độ khó và ngôn ngữ lập trình.

#### 1.1.4.2 Trong nước

- **VNOJ (VNOI Online Judge)** — Online judge **chính thức** của cộng đồng VNOI, phát triển dựa trên DMOJ; tập trung cung cấp **môi trường luyện tập & thi đấu** cho học sinh/sinh viên Việt Nam, lưu trữ các bộ đề VOI/ICPC trong nước, có hệ thống contest định kỳ.
- **MarisaOJ** — Nền tảng luyện tập được xây dựng bởi cộng đồng cựu học sinh chuyên ở Việt Nam, có **lộ trình học (roadmap)** định hướng từ cơ bản đến nâng cao, tài liệu hướng dẫn sử dụng, và tài nguyên ghi chú giúp người mới bắt đầu **tiếp cận CP có hệ thống**.
- **LQDOJ (Lê Quý Đôn Online Judge)** — Online judge phát triển trên nền tảng DMOJ, khởi nguồn phục vụ học sinh Trường THPT Chuyên Lê Quý Đôn và **mở đăng ký công khai** cho cộng đồng; có bảng xếp hạng, chấm tự động, và contest nội bộ/định kỳ.

<sup>1</sup>Quy tắc chấm thời gian và phạt được mô tả trong quy định World Finals/Regionals.

<sup>2</sup>Xem quy định eligibility của khu vực châu Á và ICPC Regionals.

## 1.2 Ngôn ngữ lập trình trong CP

1.2.1 Vì sao C++ là lựa chọn hàng đầu?

1.2.2 So sánh nhanh với Python, Java

## 1.3 Cài đặt môi trường

1.3.1 Cài đặt g++/MinGW, VS Code

1.3.2 Thiết lập template C++ cho CP

## 1.4 Giới thiệu các nền tảng luyện tập

# CHƯƠNG 2

## GIỚI THIỆU VỀ LẬP TRÌNH

# CHƯƠNG 3

## GIỚI THIỆU VỀ LẬP TRÌNH



# CHƯƠNG 4

## GIỚI THIỆU VỀ LẬP TRÌNH

# CHƯƠNG 5

## GIỚI THIỆU VỀ LẬP TRÌNH

# **CHƯƠNG 6**

## **GIỚI THIỆU VỀ LẬP TRÌNH**

# **CHƯƠNG 7**

## **GIỚI THIỆU VỀ LẬP TRÌNH**

# **CHƯƠNG 8**

## **GIỚI THIỆU VỀ LẬP TRÌNH**

# **CHƯƠNG 9**

## **GIỚI THIỆU VỀ LẬP TRÌNH**

# **CHƯƠNG 10**

## **GIỚI THIỆU VỀ LẬP TRÌNH**

