

W01 - 2021-10-06 - Intro - Notes

tags: NMLT-21CTT2-HTThanh-2021

Đa số tài liệu của IT là tiếng Anh, cố gắng tập luyện đọc tài liệu tiếng Anh.

Khi làm 1 bài tập:

- Đọc hiểu đề bài, hỏi bạn bè, thầy cô, tìm ra giải thuật, code C/C++.
- Đọc hiểu đề bài, hỏi bạn bè, thầy cô, tham khảo bài giải trên Internet, tắt bài giải đó đi, tự code lại theo những gì mình hiểu, mình nhớ. Nếu bạn thấy sau khi tham khảo và làm, bài làm của mình giống quá nhiều so với Internet, đừng nộp bài đó.

Khuyến khích học nhóm, chỉ bài lẫn nhau trong lớp:

- Chỉ được chỉ nhau về giải thuật.
- Không share code.

Khác biệt giữa THPT, Đại học:

- GV sẽ ko nhớ tên của tất cả mọi người.
- Nhiệm vụ của SV, phải tạo ấn tượng tốt với GV (tranh thủ phát biểu, giới thiệu tên với GV).
- Đặt mục tiêu: mỗi môn học, tôi phát biểu ít nhất 1 lần.

Khi cài Visual Studio Community: nhớ check full option (hoặc ít nhất là check full option C++)

Programmer, Developer, Coder (code, đui, đơ)

Lập trình là một nghệ thuật. Người lập trình là một nghệ sĩ.

Lập trình là cài đặt một hay nhiều thuật toán. Suy ra, để lập trình được, thì trước tiên phải có thuật toán.

Lập trình là cài đặt bài một ngôn ngữ lập trình (Pascal, C/C++, C#, Java, Swift, Javascript, Python, Ruby, Golang).

Lập trình là để tạo ra 1 chương trình, đem chương trình đó đi bán cho khách hàng. Chương trình có thể trên máy tính, một website, chương trình trên Android, trên iOS, trên thiết bị nhúng.

Trong chương trình học, sẽ ko có môn Nhập môn lập trình Java, Nhập môn lập trình C#, Nhập môn lập trình Javascript, Nhập môn lập trình Python, Nhập môn lập trình Golang... bởi vì đã có môn Nhập môn lập trình rồi.

Thuật toán là một tập hợp các bước. Suy ra, 1 thuật toán thường có nhiều bước.

Thuật toán là một tập hợp các bước hữu hạn. Suy ra, khi viết 1 thuật toán, ta phải biết thuật toán đó có bao nhiêu bước? (5 bước, 50 bước)

Thuật toán là phải định nghĩa rõ ràng.

Thuật toán là các bước để giải quyết 1 bài toán.

Lập trình là cài đặt một hay nhiều thuật toán. Thuật toán là cái có trước, code là cái có sau.

Tìm SĐT của một người trong Cuốn danh bạ điện thoại.

Input: tên của người cần tìm.

Output: số điện thoại của người cần tìm hoặc thông báo là ko tìm thấy.

Thuật toán Tìm SĐT của một người trong cuốn danh bạ điện thoại - Thuật toán 1

Tìm từ đầu danh bạ đến khi tìm thấy tên của ông John, mỗi lật 1 trang.

B1: Lật 1 trang, lật trang thứ i .

B2: Tìm tên người đó ở trong trang thứ i .

B3: Nếu tìm thấy tên người đó ở trong trang thứ i , ghi nhận lại SĐT. Kết thúc quá trình tìm kiếm.

B4: Nếu ko tìm thấy tên người đó ở trong trang thứ i và trang thứ i là trang cuối cùng, thông báo ko tìm thấy SĐT của người cần tìm.

B5: Nếu ko tìm thấy tên người đó ở trong trang thứ i và chưa đến trang cuối, quay lại B1.

Thuật toán Tìm SĐT của một người trong cuốn danh bạ điện thoại - Thuật toán 2

B1: Lật giữa cuốn danh bạ, giả sử trang đó là trang thứ i .

B2: Tìm tên người đó ở trong trang thứ i .

B3: Nếu tìm thấy tên người đó ở trong trang thứ i , ghi nhận lại SĐT. Kết thúc quá trình tìm kiếm.

B4: Nếu ko tìm thấy tên người đó ở trong trang thứ i , kiểm tra tên của người đó sẽ nằm ở nửa đầu vs nửa sau của cuốn danh bạ (tên trong cuốn danh bạ được sắp xếp ABCD..Z).

B5: Lặp lại bước 1 với 1 nửa đầu hoặc 1 nửa sau.

B6: Nếu ko tìm thấy tên người đó ở trong trang thứ i và nửa đầu hoặc nửa sau cần xét ko còn trang nào cả, thông báo ko tìm thấy SĐT của người cần tìm.

Cùng 1 bài toán (vấn đề), sẽ có nhiều thuật toán (đúng), nhanh chậm khác nhau.

Sau này, khi lập trình, cẩn thận Ctrl-C, Ctrl-V, sửa đổi lại cho đúng.

Thuật toán là do con người (lập trình viên) nghĩ ra. Lập trình là con người chỉ cho máy tính cách làm việc. Suy ra, con người thông minh hơn máy tính. Nếu thuật toán sai (code sai), máy tính ko bao giờ sai, chỉ có lập trình viên sai.

Sai lầm: thầy ơi, em ko biết lập trình. Thật ra, nhiều khả năng, bạn đó chưa nghĩ ra thuật toán.

Các bước lập trình:

- Đọc hiểu đề bài, input là cái gì? output là cái gì? => Nếu ko hiểu, đọc lại, đi hỏi bạn bè, giáo viên, khách hàng. (cẩn thận, hiểu sai, làm sai, 0đ trong trường, mất tiền ở ngoài đời)
- Lựa chọn phương pháp.
- Đưa ra ý tưởng về cách làm, xây dựng thuật toán. Nếu ko ý tưởng gì cả, đưa ra ý tưởng sơ khởi cho bạn bè tham khảo, hỏi bạn bè, giáo viên, internet. Nếu bạn có thuật toán rồi, bạn thấy có lỗi hổng, bạn có thể share thuật toán đó cho mọi người, nhận góp ý.
- Cài đặt chương trình bằng 1 ngôn ngữ lập trình nào đó.
- Chương trình bạn viết ra, 90% là sẽ có lỗi (lỗi lớn, lỗi nhỏ), hiệu chỉnh chương trình, hỏi bạn bè, thầy cô.
- Thực hiện chương trình.
- Đóng gói chương trình, bàn giao khách hàng, lấy tiền.

Xây dựng thuật toán (B1, B2...): ngôn ngữ tự nhiên hoặc sơ đồ khối

Bài 1001: Nhập bán kính của đường tròn. Tính chu vi và diện tích của hình tròn đó.

B1: Nhập vào bán kính r (Hỏi người dùng, r bằng bao nhiêu)

B2: Tính chu vi theo công thức, $C = 2 * \pi * r$.

B3: Tính diện tích theo công thức, $S = \pi * r * r$

B4: In kết quả: chu vi, diện tích

B1: Nhập vào bán kính r (Hỏi người dùng, r bằng bao nhiêu)

B2: Tính chu vi theo công thức, $C = 2 * \pi * r$. In chu vi.

B3: Tính diện tích theo công thức, $S = \pi * r * r$. In diện tích.

Một chương trình, thường có 3 giai đoạn:

- Nhập dữ liệu (B1)
- Xử lý, tính toán (B2, B3)
- In kết quả (B4)

Bài 0004:

$$\begin{aligned} S &= 1/(2^1) + 1/(2^2) + 1/(2^3) + 1/(2^4) + 1/(2^5) \\ &= 1/2 + 1/4 + 1/6 + 1/8 + 1/10 \\ &= 1.141666667 \end{aligned}$$

Điều kiện: $n \geq 1$

Chạy từng bước sơ đồ khối.

Ý tưởng làm các bài này là tính tổng DẪN DẪN, tích tiểu thành đại.

$$n = 5$$

$$S = 0$$

$$i = 1$$

$$1 \leq 5$$

$$S = S + 1/2^i = 0 + 1/(2^1) = 1/2$$

$$i = 2$$

$$2 \leq 5$$

$$S = S + 1/2^i = 1/2 + 1/(2^2) = 1/2 + 1/4$$

$$i = 3$$

$$3 \leq 5$$

$$S = S + 1/2^i = 1/2 + 1/4 + 1/6$$

$$i = 4$$

$$4 \leq 5$$

$$S = S + 1/2^i = 1/2 + 1/4 + 1/6 + 1/8$$

$$i = 5$$

$$5 \leq 5$$

$$S = S + 1/2^i = 1/2 + 1/4 + 1/6 + 1/8 + 1/10$$

$$i = 6$$

6 <= 5: False

ln S = 1.14