***Câu 1:*** *Trong khoa học máy tính,* ***cấu trúc dữ liệu*** *được hiểu như thế nào? Cho ví dụ.*

Trong [khoa học máy tính](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh), **cấu trúc dữ liệu** là một cách lưu [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u) trong [máy tính](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh) sao cho nó có thể được sử dụng một cách hiệu quả.[[1]](https://vi.wikipedia.org/wiki/C%E1%BA%A5u_tr%C3%BAc_d%E1%BB%AF_li%E1%BB%87u#cite_note-1)[[2]](https://vi.wikipedia.org/wiki/C%E1%BA%A5u_tr%C3%BAc_d%E1%BB%AF_li%E1%BB%87u#cite_note-2)

 Ví dụ thư viện mẫu chuẩn của C++, [Java](https://vi.wikipedia.org/wiki/Java) API, và Microsoft [.NET Framework](https://vi.wikipedia.org/wiki/.NET_Framework).

***Câu 2:*** *Trong khoa học máy tính,* ***giải thuật*** *được hiểu như thế nào? Cho ví dụ.*

Trong [toán học](https://vi.wikipedia.org/wiki/To%C3%A1n_h%E1%BB%8Dc) và [khoa học máy tính](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh), một **thuật toán**, còn gọi là **giải thuật**, là một [tập hợp hữu hạn](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_h%E1%BB%A3p#L%E1%BB%B1c_l%C6%B0%E1%BB%A3ng_c%E1%BB%A7a_t%E1%BA%ADp_h%E1%BB%A3p_-_H%E1%BB%AFu_h%E1%BA%A1n_v%C3%A0_v%C3%B4_h%E1%BA%A1n) các hướng dẫn [được xác định rõ ràng](https://vi.wikipedia.org/w/index.php?title=%C4%90%C6%B0%E1%BB%A3c_x%C3%A1c_%C4%91%E1%BB%8Bnh_r%C3%B5&action=edit&redlink=1), có thể thực hiện được bằng máy tính, thường để giải quyết một lớp vấn đề hoặc để thực hiện một phép tính.[[1]](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n#cite_note-1)[[2]](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n#cite_note-2) Các thuật toán luôn [rõ ràng](https://vi.wikipedia.org/w/index.php?title=T%E1%BB%91i_ngh%C4%A9a&action=edit&redlink=1) và được sử dụng chỉ rõ việc thực hiện các [phép tính](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A9p_t%C3%ADnh&action=edit&redlink=1), [xử lý dữ liệu](https://vi.wikipedia.org/wiki/X%E1%BB%AD_l%C3%AD_d%E1%BB%AF_li%E1%BB%87u), [suy luận tự động](https://vi.wikipedia.org/w/index.php?title=Suy_lu%E1%BA%ADn_t%E1%BB%B1_%C4%91%E1%BB%99ng&action=edit&redlink=1) và các tác vụ khác.

Ví dụ: Lập chương trình nhập vào tọa độ các đỉnh của 1 tam giác bất kỳ trong mặt phẳng. Tính diện tích và chu vi của tam giác đó. In kết quả lên màn hình Với bài toán trên bạn sẽ lập từng bước như thế nào để giải bài toán?

***Câu 3:*** *Tại sao nói CTDL và GT có quan hệ mật thiết với nhau? Liệt kê 1 ví dụ nói về cách thiết kế cấu trúc dữ liệu sẽ ảnh hưởng đến giải thuật, giải thích tại sao?*

* Xét tới giải thuật thì phải xét giải thuật đó tác động trên cấu trúc dữ liệu nào. l Xét tới cấu trúc dữ liệu thì phải hiểu cấu trúc dữ liệu đó cần được tác động bằng giải thuật gì để được kết quả mong muốn. l Cấu trúc dữ liệu nào thì giải thuật đó. Khi cấu trúc dữ liệu thay đổi giải thuật cũng thay đổi theo. Mối quan hệ giữa cấu trúc dữ liệu và giải thuật được Niklaus Wirth tổng kết như sau: Cấu trúc dữ liệu + Giải thuật = Chương trình
* **Ví dụ:** Dùng ngôn ngữ viết để tìm ra 3 số lớn nhất trong a, b, c

+ Bước 1: Gán max = a.

+ Bước 2: Nếu b > max thì gán max = b.

+ Bước 3: Nếu c > max thì max chính là c.

Thiết kế giải thuật bằng ngôn ngữ viết là **liệt kê tuần tự các bước** bằng ngôn ngữ tự nhiên để biểu diễn thuật toán.

**+ Ưu điểm:**

Đơn giản, không cần kiến thức về cách biểu diễn (mã giả, lưu đồ,...)

**+ Nhược điểm:**

* Dài dòng, không có cấu trúc.
* Đôi lúc khó hiểu và không biểu diễn được thuật toán.

**Câu 4:** *Đếm số* ***phép so sánh*** *trong giải thuật ở ví dụ 1.12.*

*void InsertionSort(int a[], int n)*

*{*

*int i, j, x;*

*for (i = 1; i < n; i++)*

*{*

*x = a[i]; j=i;*

*while (j > 0 && a[j-1] > x)*

*{*

*a[j] = a[j-1];*

*j- -;*

*}*

*a[j] = x;*

*}*

*}*

* + Số phép so sánh: 2n(n-1)