

So sánh Danh sách liên kết và Mảng

REB đăng lúc 2:59 CH 03/06/2018

Vì vậy, hôm nay mình sẽ so sánh, cũng như nói về sự khác nhau, ưu, nhược điểm của danh sách liên kết so với mảng để các bạn có thể hiểu rõ hơn về chúng.

Danh sách liên kết và mảng là hai kiểu cấu trúc dữ liệu rất phổ biến trong lập trình, đặc biệt với những ai đang học hay tìm hiểu về Cấu trúc dữ liệu và giải thuật có lẽ đều đã nghe về chúng. Vì vậy, hôm nay mình sẽ so sánh, cũng như nói về sự khác nhau, ưu, nhược điểm của danh sách liên kết so với mảng để các bạn có thể hiểu rõ hơn về chúng.

So sánh Danh sách liên kết và Mảng

Rebvn.com

1. Mảng là gì?

Mảng là một tập hợp các phần tử cố định có cùng một kiểu, được lưu trữ liên tiếp nhau trong các ô nhớ. Kiểu phần tử có thể là có các kiểu bất kỳ: ký tự, số, chuỗi ký tự...; cũng có khi ta sử dụng kiểu mảng để làm kiểu phần tử cho một mảng (trong trường hợp này ta gọi là mảng của mảng hay mảng nhiều chiều). (- Theo *Wikipedia*)

Các phần tử của mảng được lưu trữ trong các ô nhớ nằm liên tục nhau.

Việc khai báo và sử dụng mảng thì khá đơn giản, và cũng có nhiều trên Google rồi nên mình sẽ không nói ở đây.

2. Danh sách liên kết là gì?

Danh sách liên kết có thể hiểu đơn giản là một cấu trúc dữ liệu gồm các nút được kết nối với nhau thông qua các liên kết (thường là con trỏ) tới nút khác trong danh sách. Một nút trong danh sách liên sẽ có cấu tạo như sau :



+ Sẽ chứa thông tin mà người dùng cần.

Ví dụ : ta cần một danh sách liên kết lưu các số tự nhiên từ 1 đến 10, thì thông tin mà mỗi nút trong danh sách liên kết lưu trữ là một trong các số tự nhiên từ 1 đến 10.)

+ Chứa liên kết tới nút khác trong cùng danh sách. Liên kết này có tác dụng cho máy tính biết những nút nào là trong cùng một danh sách. Mỗi nút thường chỉ được liên kết tới các nút lân cận nó chứ không phải tới tất cả các nút trong danh sách. Liên kết sử dụng thường là liên kết con trỏ.

Ví dụ : Khi xếp hàng, mỗi người chỉ cần nhớ người đứng trước mình trong hàng là ai, thì những lần sau ta đã có thể xếp lại hàng với vị trí đứng như cũ chứ không cần thiết phải nhớ hết tất cả thứ tự của hàng để xếp lại. (Người đầu hàng sẽ phải nhớ mình là người đầu hàng.) Danh sách liên kết cũng vậy, mỗi nút chỉ cần có liên kết tới các nút lân cận để có thể hoàn thiện danh sách.

Ở bài này, mình sẽ không nói về cách khai báo, tạo một danh sách liên kết. Mà chỉ tập trung vào sự khác nhau cũng như ưu, nhược điểm của danh sách liên kết so với mảng.

3. Sự khác nhau (ưu, nhược điểm) của Danh sách liên kết (DSLK) và Mảng.

Danh sách liên kết và Mảng là hai kiểu cấu trúc dữ liệu rất phổ biến, và chúng có tính liên quan không hề nhẹ. Cụ thể là ưu điểm của thằng này chính là nhược của thằng kia và nhược của thằng kia lại chính là ưu của thằng này. Cũng như việc không có ai hoàn hảo cả, vì thế mà tùy tình huống ta sẽ chọn sử dụng thằng nào mà bỏ thằng còn lại. (Các bạn đừng liên tưởng tới việc Crush bỏ bạn để đi theo một thằng nhà giàu, đẹp zai khác nhưng lãng nhăng nhé. 😊😊😊 Mình xin lỗi nếu có "vô tình" làm bạn nào đó buồn. Mình cũng FA chứ có hơn gì các bạn đâu :((()



Quay lại vấn đề chính :

- Ưu điểm của Mảng, nhược điểm của DSLK :

+ Vì DSLK cần lưu trữ hai thứ đó là thông tin (dữ liệu) của nút đó và liên kết tới nút khác trong danh sách nên sẽ tốn bộ nhớ hơn so với mảng. Mảng thì chỉ cần một ô nhớ và lưu dữ liệu vào ô nhớ đó thôi.

+ Truy xuất tới một vị trí trong mảng sẽ nhanh hơn, ta chỉ cần gọi Mảng[vị trí cần truy xuất] là được rồi. Còn với danh sách liên kết, khi cần truy xuất tới một nút cụ thể nào đó, ta sẽ cần đi từ đầu danh sách hoặc cuối danh sách để tới được vị trí nút đó. Tốn thời gian hơn so với mảng.

+ Ngoài ra, DSLK còn một nhược điểm nữa là nếu một nút trong danh sách bị hỏng, thì cả danh sách sẽ bị hỏng theo vì ta sẽ không thể truy xuất tới một số phần tử trong danh sách nữa. DSLK thường được chúng ta tự tạo và định nghĩa cho máy tính, vì thế trong quá trình tạo một DSLK có thể nhầm lẫn dẫn đến hỏng cả danh sách. Còn mảng thì thường được hỗ trợ sẵn trong ngôn ngữ lập trình mà ta sử dụng, mà mấy cái này toàn do mấy bác IQ 200 làm nên việc sai sót là ít hơn nếu không muốn nói sai thì là do ta không biết dùng 😊.

Nói về ưu điểm của Mảng nhiều quá, hơi tội cho DSLK rồi 😊. Nên tiếp theo sẽ là:

- Ưu điểm của DSLK và nhược điểm của Mảng :

+ Ở mảng, các phần tử cần được lưu trữ trong các ô nhớ nằm liên tục nhau, dẫn đến tình trạng, đôi khi ta cần cấp phát 10 ô nhớ cho mảng này, nhưng bộ nhớ lại trống chỗ 5 ô, chỗ kia 5 ô. Nói chung là 5 ô trống này không liên tục với 5 ô trống còn lại, nên không thể cấp phát được.

Lúc này, ta có thể dùng DSLK, vì DSLK gồm các nút liên kết với nhau (thường là con trỏ) nên ta có thể xếp các nút ở bất kì vị trí nào trong bộ nhớ mà không cần xếp liên tục. Ví dụ : nút A lưu ở ô nhớ có địa chỉ 01ABC, nút B lưu ở ô nhớ có địa chỉ 02XYZ, thì nút A chỉ cần có liên kết tới địa chỉ nút B để cho biết nút sau nó là nút nào, là đủ để biết toàn bộ danh sách gồm những nút nào rồi. Vì thế mà 2 nút không cần phải cạnh nhau vẫn lưu trữ được.

Điều này giúp khắc phục tình trạng phân mảnh bộ nhớ của máy tính.

+ Nếu các bạn từng làm các bài tập về Mảng như thêm, xóa phần tử, sẽ thấy việc thêm hay xóa phần tử trong mảng sẽ "khá phức tạp" và tốn công, ta sẽ phải di chuyển vị trí của các phần tử khác phần tử cần thêm (xóa) trong mảng. Còn với DSLK, ta chỉ cần bề nổi một liên kết là xong.

Ví dụ : thêm nút C vào giữa nút A và nút B thì ta chỉ cần bề liên kết từ A tới B sang từ A tới C, và cho C liên kết tới B là đã được A C B rồi. Không cần quan tâm tới các nút D hay E hay gì cả.

Từ A --> B thành A --> C và cho thêm C --> B là đã được A --> C --> B.

Ngoài ra, DLSK sẽ không cần khai báo số phần tử như mảng tĩnh, còn mảng động thì cũng không cần khai báo rồi rôi. (._.!).

Hết rồi (hoặc còn mà do kiến thức hạn hẹp nên mình không biết ._.). Do đó, mỗi cái có một ưu nhược điểm riêng mà tùy vào mục đích, trường hợp hay đôi khi đơn giản là do mình thích thôi mà các bạn có thể chọn Danh sách liên kết hay Mảng cho phù hợp với bài toán cần giải quyết.

Nếu có thắc mắc, hay trợ giúp hay góp ý gì, các bạn hãy comment dưới bài viết này nhé. Cảm ơn các bạn đã đọc bài.

Nguồn : Tham khảo đa nguồn trên internet.

Tổng hợp bởi [Rebvn.com](#). Bản quyền bài viết này thuộc về [Rebvn.com](#).

< Bài đăng mới hơn

Bài đăng cũ hơn >

👍 Thích 0

🔗 Chia sẻ



0 bình luận

Sắp xếp theo

Cũ nhất ⌵



Thêm bình luận...



📘 Plugin bình luận trên Facebook

Tags: [Coding](#) [Thuật toán](#)

BẠN CÓ THỂ QUAN TÂM

Tổng hợp các packages và cài đặt cần thiết cho việc lập trình Python trên Sublime Text

Cây nhị phân tìm kiếm trong C và CSharp
THUẬT TOÁN 13:30 25/10/2018

Danh sách liên kết đôi trong CSharp
THUẬT TOÁN 15:44 15/10/2018

Khởi tạo và thêm phần tử vào List Struct trong CSharp

CODING 17:18 05/09/2018

Python cho người mới bắt đầu : cài đặt và tài liệu

CODING 21:11 05/07/2018

Danh sách liên kết trong CSharp

THUẬT TOÁN 13:01 21/06/2018

Hiển thị thêm ▾

