- Bất cứ ai trong lúc tối ưu code ruby luôn nhập tâm nằm lòng các nguyên tắc:

* Tìm kiếm trên hash luôn nhanh hơn tìm kiếm trên array
* Tránh các vòng lặp lồng nhau
* Hạn chế query database khi hiển thị một list trên view

- O lớn là ký hiệu mô tả độ phức tạp của một thuật toán, hay nói đơn giản là performace của code trên một database bất kỳ. O lớn đại diện cho trường hợp xấu nhất có thể sảy ra, hay cận trên của thuật toán. Cùng với O lớn, còn có các ký hiệu Ω(omega) và Θ(teta) đại diện cho trường hợp tốt nhất và trường hợp trung bình.

- Performance thường được hiểu là khái niệm bao gồm hai tiêu chí: **tốc độ xử lý và bộ nhớ cần sử dụng.** Trong ngành khoa học máy tính hai khái niệm này thường được gọi là **"độ phức tạp thời gian" và "độ phức tạp không gian".** Khái niệm O lớn có thể dùng cho cả hai loại độ phức tạp trên, **nhưng với độ lớn gần như vô hạn của phần cứng bộ nhớ và mục tiêu chính của chúng ta thường là cải thiện tốc độ, nên O lớn thường được dùng với khái niệm thời gian.**

Với những bài toán nhỏ, O lớn không ảnh hưởng nhiều đến thời gian tính toán. Nhưng khi lượng dữ liệu tăng dần, O lớn sẽ quyết định tốc độ xử lý tăng tương ứng bao nhiêu lần.

- **Nói một cách đơn giản hơn, độ phức tạp là một phép tính gần đúng của số bước cần thiết để thực hiện một thuật toán.** Khi chúng tôi đánh giá độ phức tạp, chúng tôi nói về thứ tự hoạt động, không phải về số lượng chính xác của chúng. Ví dụ, nếu chúng ta có một trật tự của N2 hoạt động để xử lý các yếu tố N, sau đó N2 /2 và 3 \* N2 là số một và thứ tự bậc hai tương tự.

- Độ phức tạp của thuật toán thường được biểu diễn bằng ký hiệu O (f) , còn được gọi là ký hiệu tiệm cận hoặc “ ký hiệu O lớn ”, trong đó f là hàm của kích thước của dữ liệu đầu vào. Độ phức tạp tính toán tiệm cận O (f) đo thứ tự của các tài nguyên được tiêu thụ (thời gian CPU, bộ nhớ, v.v.) bằng một thuật toán nhất định được biểu thị dưới dạng hàm của kích thước dữ liệu đầu vào.

-  Giả sử ta tiến hành đếm như vậy với tất cả các bộ dữ liệu có kích thước tối đa N. Ký hiệu f(N) là số bước cần thực hiện nhiều nhất (trường hợp xấu nhất). Hàm f được gọi là độ phức tạp thời gian, hoặc ngắn gọn thời gian chạy của thuật toán.

|  |  |
| --- | --- |
|  | O(n!) |
| Tìm tất cả các số nhị phân | O(2n) |
| 2 vòng for() | O(n2) |
| sort | O(nlog(n)) |
| for() | O(n) |
| Tìm kiếm nhị phân (BS) | O(log(n)) |
| a = b + c; | O(1) |