**3.4. HIỆU LỰC CỦA ĐỆ QUY**

Qua các ví dụ trên, ta có thể thấy đệ quy là một công cụ mạnh để giải các bài toán. Có những bài toán mà bên cạnh giải thuật đệ quy vẫn có những giải thuật lặp khá đơn giản và hữu hiệu. Chẳng hạn bài toán tính giai thừa hay tính số Fibonacci. Tuy vậy, đệ quy vẫn có vai trò xứng đáng của nó, có nhiều bài toán mà việc thiết kế giải thuật đệ quy đơn giản hơn nhiều so với lời giải lặp và trong một số trường hợp chương trình đệ quy hoạt động nhanh hơn chương trình viết không có đệ quy. Giải thuật cho bài Tháp Hà Nội và thuật toán sắp xếp kiểu phân đoạn (QuickSort) mà ta sẽ nói tới trong các bài sau là những ví dụ.

Có một mối quan hệ khăng khít giữa đệ quy và quy nạp toán học. Cách giải đệ quy cho một bài toán dựa trên việc định rõ lời giải cho trường hợp suy biến (neo) rồi thiết kế làm sao để lời giải của bài toán được suy ra từ lời giải của bài toán nhỏ hơn cùng loại như thế. Tương tự như vậy, quy nạp toán học chứng minh một tính chất nào đó ứng với số tự nhiên cũng bằng cách chứng minh tính chất đó đúng với một số trường hợp cơ sở (thường người ta chứng minh nó đúng với 0 hay đúng với 1) và sau đó chứng minh tính chất đó sẽ đúng với n bất kỳ nếu nó đã đúng với mọi số tự nhiên nhỏ hơn n.

Do đó ta không lấy làm ngạc nhiên khi thấy quy nạp toán học được dùng để chứng minh các tính chất có liên quan tới giải thuật đệ quy. Chẳng hạn: Chứng minh số phép chuyển đĩa để giải bài toán Tháp Hà Nội với n đĩa là 2n-1:

Rõ ràng là tính chất này đúng với n = 1, bởi ta cần 21 - 1 = 1 lần chuyển đĩa để thực hiện yêu cầu

Với n > 1; Giả sử rằng để chuyển n - 1 đĩa giữa hai cọc ta cần 2n-1 - 1 phép chuyển đĩa, khi đó để chuyển n đĩa từ cọc x sang cọc y, nhìn vào giải thuật đệ quy ta có thể thấy rằng trong trường hợp này nó cần (2n-1 - 1) + 1 + (2n-1 - 1) = 2n - 1 phép chuyển đĩa. Tính chất được chứng minh đúng với n

Vậy thì công thức này sẽ đúng với mọi n.

Thật đáng tiếc nếu như chúng ta phải lập trình với một công cụ không cho phép đệ quy, nhưng như vậy không có nghĩa là ta bó tay trước một bài toán mang tính đệ quy. Mọi giải thuật đệ quy đều có cách thay thế bằng một giải thuật không đệ quy (khử đệ quy), có thể nói được như vậy bởi tất cả các chương trình con đệ quy sẽ đều được trình dịch chuyển thành những mã lệnh không đệ quy trước khi giao cho máy tính thực hiện.

Việc tìm hiểu cách khử đệ quy một cách "máy móc" như các chương trình dịch thì chỉ cần hiểu rõ cơ chế xếp chồng của các thủ tục trong một dây chuyền gọi đệ quy là có thể làm được. Nhưng muốn khử đệ quy một cách tinh tế thì phải tuỳ thuộc vào từng bài toán mà khử đệ quy cho khéo. Không phải tìm đâu xa, những kỹ thuật giải công thức truy hồi bằng quy hoạch động là ví dụ cho thấy tính nghệ thuật trong những cách tiếp cận bài toán mang bản chất đệ quy để tìm ra một giải thuật không đệ quy đầy hiệu quả.