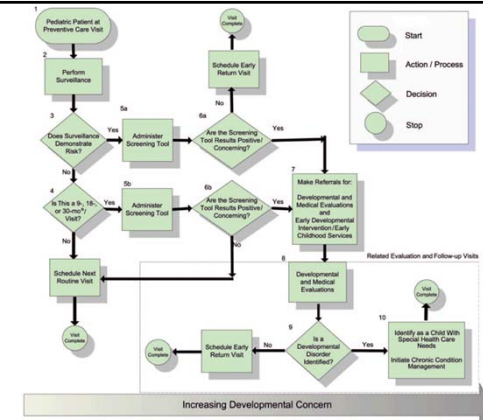


Phần I – Giới thiệu về Thuật toán



Chương 1.1

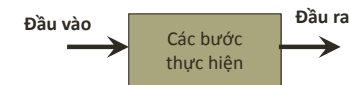
KHÁI NIỆM CƠ BẢN

Nội dung

- 1.1. Thuật toán là gì?
- 1.2 Tính chất của thuật toán
 - 1.2 .1 Tính chính xác
 - 1.2.2 Tính hiệu quả
- 1.3 Chứng minh thuật toán đúng
- 1.4 Biểu diễn thuật toán

1.1 Thuật toán là gì ?

- **Thuật toán:**
 - thủ tục để thực hiện một nhiệm vụ cụ thể
 - ý tưởng nằm sau các chương trình máy tính.
- Thuật toán phải giải quyết bài toán tổng quát, và được định nghĩa rõ ràng.
- Một thuật toán giải bài toán đặt ra là một thủ tục xác định bao gồm một **dãy hữu hạn các bước** cần thực hiện để thu được **đầu ra** cho một **đầu vào** cho trước của bài toán.



1.1 Thuật toán là gì ?



Bài toán: sắp xếp

- **Đầu vào:** một dãy gồm n khóa a_1, a_2, \dots, a_n
- **Đầu ra:** một hoán vị có thứ tự của các khóa đầu vào trong đó

$$a'_1 \leq a'_2 \leq \dots \leq a'_n$$

Trường hợp cụ thể của bài toán

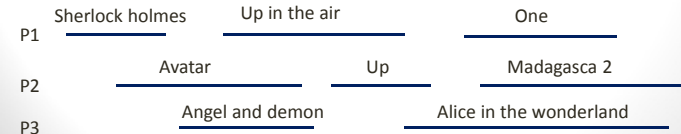
- {14, 45, 68, 24, 54, 34}
- {Mike, Bob, Sally, Jill, Jan}
- Chúng ta chỉ quan tâm đến các thuật toán **chính xác** và **hiệu quả**, và dễ cài đặt

1.2.1 Tính chính xác

- Thuật toán phải cho đầu ra mong muốn ứng với bất cứ đầu vào hợp lệ nào của bài toán.
- Tính chính xác không phải lúc nào cũng dễ thấy!

VD. Bài toán chọn lịch xem phim

- **Đầu vào:** Một tập L gồm thời gian chiếu trong ngày của n bộ phim
- **Đầu ra:** Tập con của L chứa số bộ phim lớn nhất có thể xem (không được chồng nhau về thời gian)



1.2.1 Tính chính xác

- **Thuật toán 1.** Chọn bộ phim sớm nhất trong L mà không trùng với các bộ phim đã chọn trước đó. Lặp lại cho đến khi không thể chọn thêm.



- **Thuật toán 2.** Chọn bộ phim có thời gian chiếu ngắn nhất trong L mà không trùng với các bộ phim đã chọn trước. Lặp lại cho đến khi không chọn thêm được.



1.2.1 Tính chính xác

- **Thuật toán 3. Duyệt toàn bộ:** duyệt 2^n tập con của n bộ phim trong L . Chọn ra tập con nào có số lượng phần tử lớn nhất. Đảm bảo thu được kết quả tối ưu. Thuật toán chạy rất chậm, vd $n=20$ thì số tập con là 2^{20}
- **Thuật toán 4. Thuật toán tối ưu:** sắp xếp các lịch chiếu phim theo thứ tự không giảm thời gian kết thúc. Lần lượt xem xét các phim trong danh sách đã sắp xếp, bổ sung vào danh sách xem bộ phim đang xét nếu nó không chồng lên các bộ phim đã có trong danh sách xem.
- Có những bài toán mà không tồn tại thuật toán chính xác để giải!

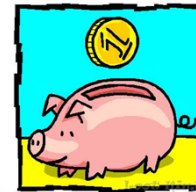
1.2.1 Tính chính xác

- **Phân biệt giữa thuật toán chính xác và không chính xác:** đưa ra một ví dụ thuật toán mà thuật toán cho kết quả sai (phản ví dụ).
- **Chứng minh tính đúng đắn của thuật toán:** khó khăn hơn nhiều.
- **Bài tập.** Tìm các phản ví dụ cho các thuật toán giải bài toán hành trình du lịch tối ưu.

1.2.2 Tính hiệu quả

"Tại sao không chỉ sử dụng mỗi siêu máy tính?"

- Siêu máy tính chỉ cho người giàu và những người quá ngốc để có thể thiết kế một thuật toán hiệu quả!
- Thuật toán nhanh hơn chạy trên các máy tính chậm hơn sẽ thắng trong trường hợp dữ liệu đầu vào đủ lớn.



Bài toán cái túi

- **Đầu vào:** n đồ vật, mỗi đồ vật i có một trọng lượng w_i và một giá trị c_i . Một cái túi có thể chứa các đồ vật với trọng lượng tối đa là b
- **Đầu ra:** Cách chắt các đồ vật vào túi sao cho trọng lượng tối đa không vượt quá b , và tổng giá trị các đồ vật trong túi là lớn nhất.

$$W = \sum_{i=1}^k w_i \leq b \quad C = \sum_{i=1}^k c_i \rightarrow \max$$

- Xây dựng thuật toán chắt các đồ vào túi ?



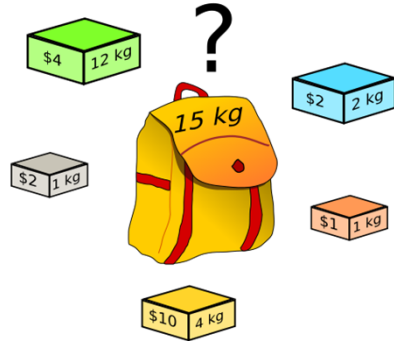
Chọn đồ vật có giá trị cao trước

- Sắp xếp các đồ vật theo thứ tự giảm về giá trị.
- Lần lượt xét các đồ theo thứ tự này, cho đồ vật đang xét vào túi nếu nó còn có thể chứa thêm được



Chọn đồ vật trọng lượng nhỏ trước

- Sắp xếp các đồ vật theo thứ tự tăng trọng lượng
- Lần lượt xét các đồ vật theo thứ tự này, chọn đồ vật đang xét vào túi nếu nó vẫn có thể chứa thêm

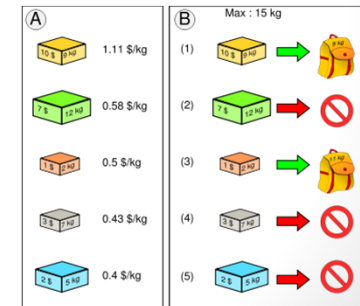


Chọn đồ vật theo tỉ lệ c_i/w_i

- Sắp xếp các đồ vật theo thứ tự giảm của tỉ lệ giá trị/ trọng lượng

$$\frac{c_1}{w_1} \geq \frac{c_2}{w_2} \geq \frac{c_3}{w_3} \dots \geq \frac{c_n}{w_n}$$

- Lần lượt xét các đồ vật theo thứ tự này, chọn đồ vật đang xét vào túi nếu nó vẫn có thể chứa thêm



Tìm phản ví dụ ?

Chứng minh thuật toán sai bằng cách chỉ ra một phản ví dụ

- Tìm trong các trường hợp dữ liệu nhỏ
- Các ví dụ mà sát với các tiêu chuẩn lựa chọn của thuật toán
- Các ví dụ của các trường hợp cực trị (lớn nhất, nhỏ nhất ...)



Không tìm được phản ví dụ không có nghĩa thuật toán là đúng!

1.3 Chứng minh tính đúng đắn

- Thuật toán được định nghĩa đệ quy:** Thuật toán được định nghĩa lại bằng chính nó (với kích thước bài toán nhỏ hơn)

$$n! = \begin{cases} 1 & \text{nếu } n = 0 \\ n \times (n-1)! & \text{nếu } n > 0 \end{cases}$$

- Chứng minh tính đúng đắn của thuật toán đệ quy bằng phương pháp quy nạp

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$



1.4 Biểu diễn thuật toán

- Cần biểu diễn các bước thực hiện tuần tự của thuật toán một cách cụ thể.

- Biểu diễn bằng:

- Ngôn ngữ tự nhiên
- Giả ngôn ngữ (pseudocode)
- Lưu đồ
- Ngôn ngữ lập trình cụ thể (C/C++, java,...)

Tính chính xác
↓



↑
Tính dễ dàng