

Phân tích & Thiết kế thuật toán (Algorithms Design & Analysis)

L/O/G/O

GV: HUỖNH THỊ THANH THƯỜNG

Email: thuonghtt@uit.edu.vn

TỔNG QUAN

CHƯƠNG 1



L/O/G/O

www.themegallery.com

Nội dung Tổng quan

1. Vấn đề và xác định vấn đề (sẽ học sau)
2. Thuật toán (nhắc lại: đã học ở môn NMLT)
3. Phân tích thuật toán
4. Độ phức tạp

Q&A

- Mục tiêu của lập trình?



Q&A

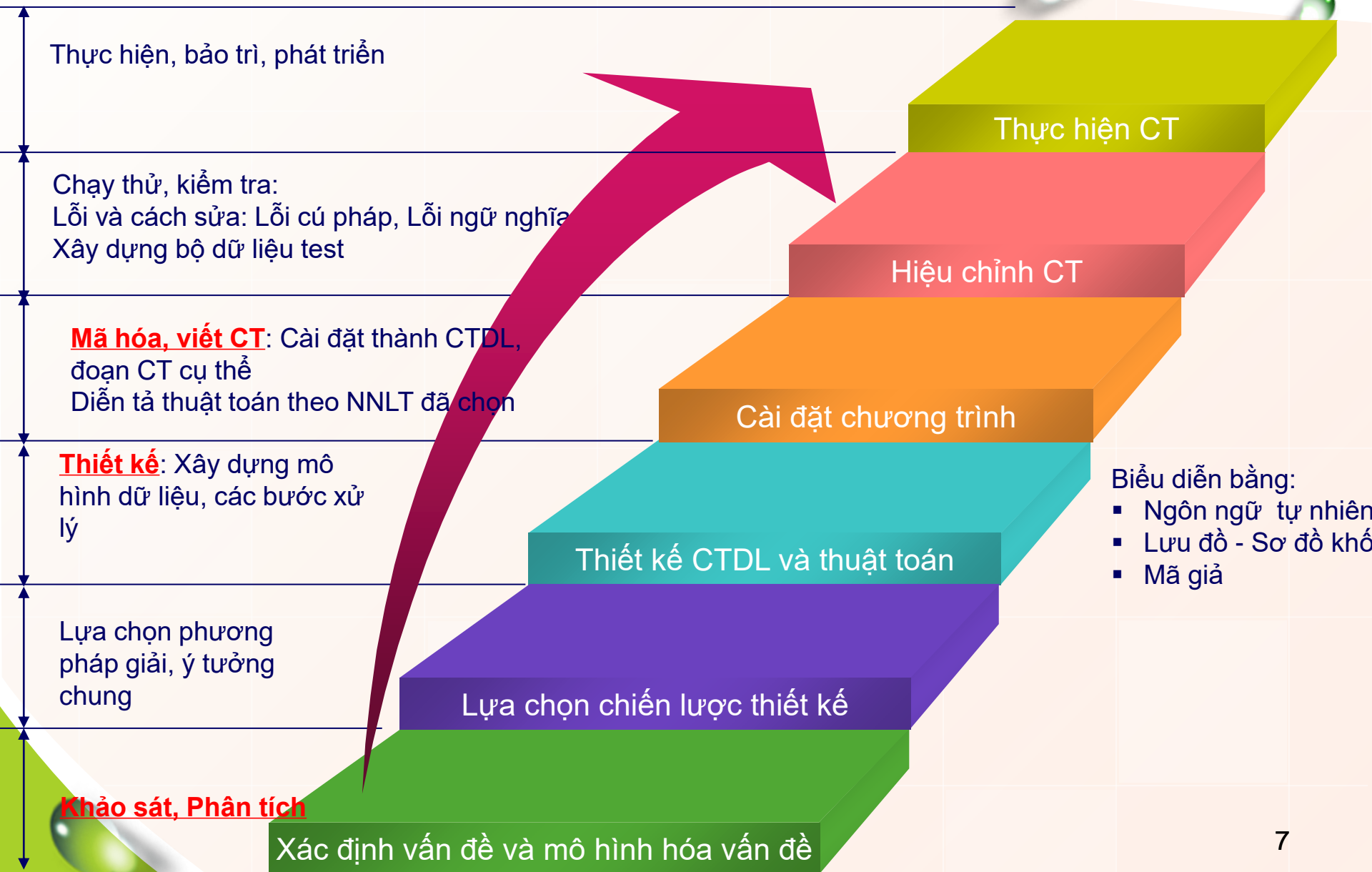
- Vấn đề , bài toán (Problem) là gì?



- Giải quyết vấn đề là làm gì?



Quy trình giải quyết vấn đề bằng MTĐT

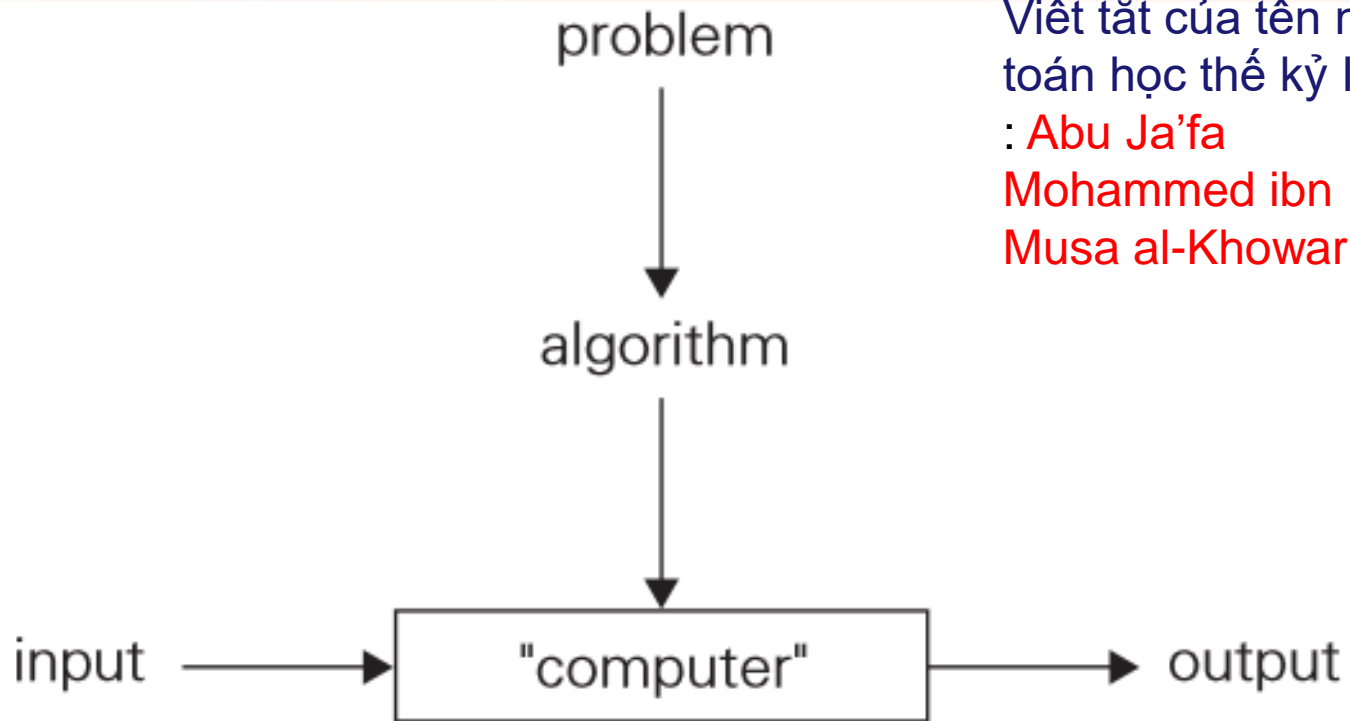


Thuật toán

- Thuật toán là gì?

Algorithm:

Viết tắt của tên nhà
toán học thế kỷ IX là
: Abu Ja'fa
Mohammed ibn
Musa al-Khowarizmi



Thuật toán là gì?

Nhắc lại: Khái niệm

“Thuật toán là một dãy hữu hạn các bước không mập mờ và có thể thực thi được, quá trình hành động theo các bước này phải dừng và cho kết quả mong muốn” (SGK)

Thuật toán là gì?

Nhắc lại: Khái niệm

- Giải pháp viết dưới dạng thủ tục
- Trình tự các bước giải quyết
- Thỏa 3 tiêu chuẩn:

Xác định + Hữu hạn + Đúng

Tìm kiếm theo chiều rộng

❖ Thuật toán:

Bước 1: Khởi tạo giá trị ban đầu cho 1 số biến điều khiển

Open := {s};

Close := {};

Bước 2: Thực hiện quá trình dò ...

While (Open \neq {})

2.1 Lấy p từ đầu Open (xoá p khỏi Open)

p <-PickHead(Open)

2.2 Nếu p là trạng thái kết thúc thì thoát, thông báo kết quả

2.3 Bỏ p vào Close

2.4 Với mỗi q kề p,

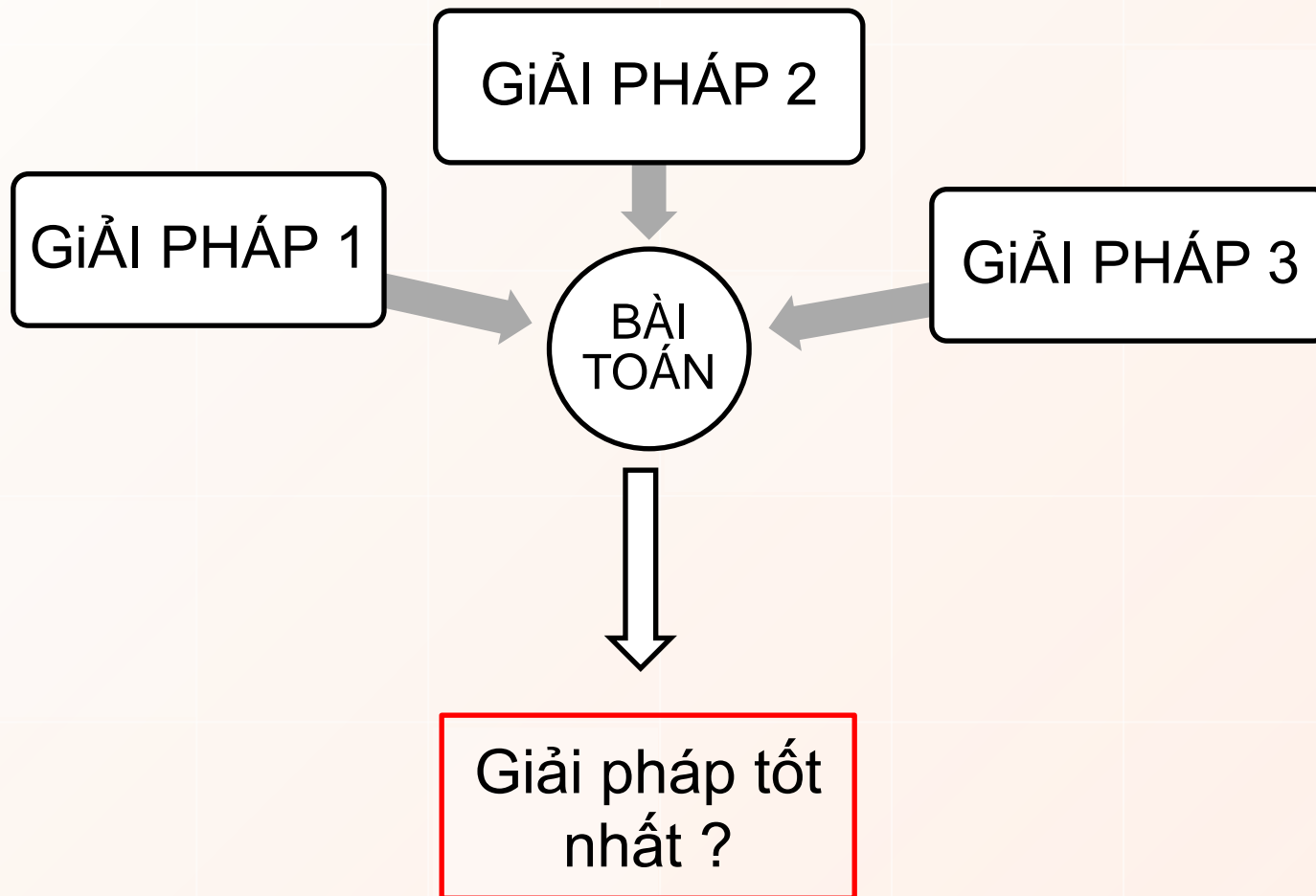
Nếu q không thuộc Close thì bỏ q vào **cuối Open**

Bước 3: Không tìm thấy kết quả

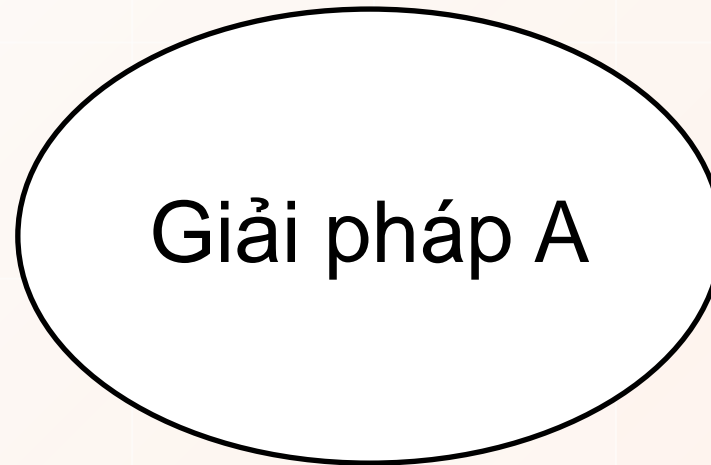
Câu hỏi thảo luận trên lớp

- 1) Tại sao phải đánh giá/phân tích giải thuật?
- 2) Đánh giá dựa trên những tiêu chuẩn nào?
- 3) Tiêu chuẩn nào quan trọng nhất?
- 4) Đánh giá bằng cách nào?

Tại sao phải phân tích?



Tại sao phải phân tích?



Liệu có thể cải tiến để tốt hơn?

Câu hỏi thảo luận trên lớp

- 1) Tại sao phải đánh giá/phân tích giải thuật?
- 2) **Đánh giá dựa trên những tiêu chuẩn nào?**
- 3) Tiêu chuẩn nào quan trọng nhất?
- 4) Đánh giá bằng cách nào?

Tính chất của thuật toán

Nhắc lại: Các tính chất

- Tính xác định/rõ ràng
- Tính đúng/chính xác
- Tính hữu hạn/dừng
- Tính đơn giản
- Tính khách quan
- Tính phổ dụng/tổng quát
- Tính hiệu quả: không gian và thời gian

Câu hỏi thảo luận trên lớp

- 1) Tại sao phải đánh giá/phân tích giải thuật?
- 2) Đánh giá dựa trên những tiêu chuẩn nào?
- 3) **Tiêu chuẩn nào quan trọng nhất?**
- 4) Đánh giá bằng cách nào?

Features of good algorithm

- Một giải thuật được xem là tốt nếu đạt được các tiêu chuẩn sau:
 - Tính đúng
 - Tính đơn giản
 - Tính tối ưu (hiệu quả)
 - Không gian (tốn ít bộ nhớ/tài nguyên máy tính)
 - Thời gian chạy của thuật toán (thực hiện nhanh)
- Trong khuôn khổ của môn học này, chúng ta chỉ quan tâm đến tiêu chuẩn **thực hiện nhanh**

Thuật giải

Giải pháp được diễn đạt tương tự như thuật toán nhưng **không đòi hỏi các tiêu chuẩn** như thuật toán.

- ✓ **Tính xác định**
- ✓ **Tính đúng:** chấp nhận các thuật giải đơn giản có thể cho kết quả **đúng hay gần đúng** nhưng có **khả năng thành công** cao hơn.

Câu hỏi thảo luận trên lớp

- 1) Tại sao phải đánh giá/phân tích giải thuật?
- 2) Đánh giá dựa trên những tiêu chuẩn nào?
- 3) Tiêu chuẩn nào quan trọng nhất?
- 4) **Đánh giá bằng cách nào?**

Analysis of Time efficiency

Measuring Running Time: Là công việc ước lượng **thời gian thực hiện của thuật toán** để so sánh tương đối các thuật toán với nhau

Làm sao ước lượng thời gian thực hiện của thuật toán?

Analysis of Time efficiency

- Hai phương pháp đánh giá tính hiệu quả về thời gian:
 - Phương pháp thực nghiệm (Thực hành).
 - Phương pháp toán học (Lý thuyết)

Empirical analysis (Thực nghiệm)

- Các bước thực hiện:
 - Viết chương trình, cài đặt
 - Chọn các bộ dữ liệu thử nghiệm
 - Thực thi chương trình với nhiều bộ dữ liệu đã chọn
 - Đo và thống kê thời gian, thông số
 - Xấp xỉ biểu đồ
- Đơn vị đo: giờ, phút, giây.
- Ưu điểm: Dễ thực hiện.
- Nhược điểm: ? (câu hỏi trên lớp)

Empirical analysis

❖ Nhược điểm:

- Chịu sự hạn chế của ngôn ngữ lập trình.
- Ảnh hưởng bởi trình độ của người lập trình.
- Phụ thuộc vào phần cứng (cấu hình máy).
- Phải chọn được các bộ dữ liệu thử đặc trưng cho tất cả tập các dữ liệu vào của thuật toán: khó khăn và tốn nhiều chi phí.
- Bộ dữ liệu không thể đặc trưng hết
- Cần phải cài đặt chương trình và đo thời gian

Mathematical analysis

- Count the number of times “each of operations” or “**basic operation**” is executed
- Basic operation: the most important operation, contribute the most to total running time

Unit for measuring running time

- Đơn vị đo thời gian thực hiện = số các lệnh được thực hiện trong một máy tính lý tưởng
- Thời gian = số phép gán + số phép so sánh (thông thường)

Ví dụ : Tính tổng các số nguyên dương từ 1 đến n

```
int Tong (int n)
{
    int S=0;
    for (int i = 1; i<=n; i++)
        S = S+i;
    return S;
}
```

Bao nhiêu câu lệnh?

Running time: Function of the input size

- ❖ Thời gian thực hiện chương trình là một hàm phụ thuộc kích thước dữ liệu vào.
- ❖ Ký hiệu: $T(n)$ với n là kích thước (độ lớn) dữ liệu vào
- ❖ $T(n) \geq 0 \quad \forall n \geq 0$
- ❖ Ví dụ: Chương trình tính tổng của n số có thời gian thực hiện là $T(n) = ?$ (đơn vị: số chỉ thị thực thi/ số phép toán)

Ví dụ

❖ Tính $T(n)$?

```
sum = 0;
i = 1;
while (i ≤ n)
{
    j = 1;
    while (j ≤ n)
    {
        sum = sum + i*j;
        j = j + 1;
    }
    i = i + 1;
}
```

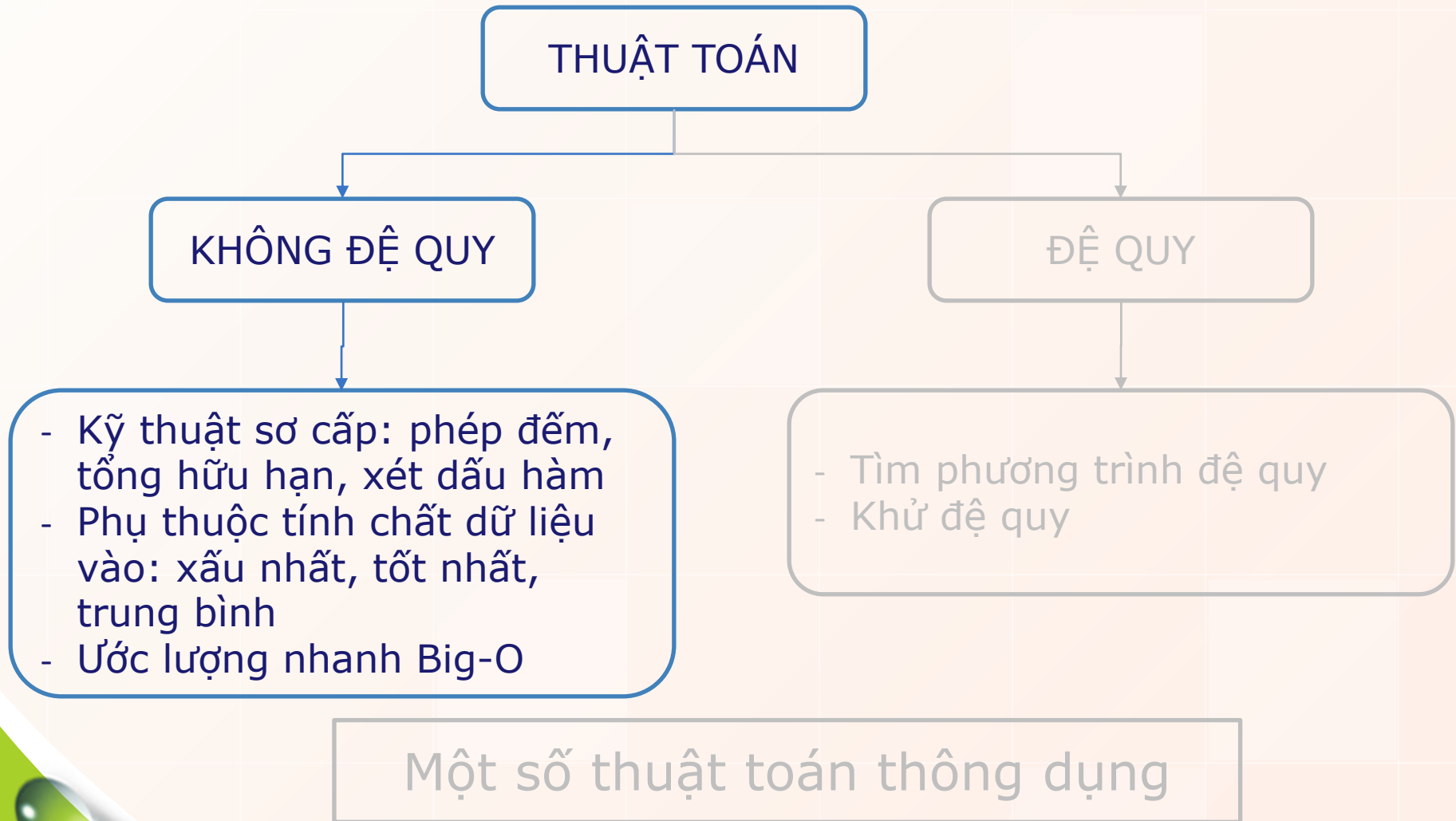
Số phép gán ?
Số phép so sánh ?
 $T(n)$?

$$T(n) = 3n^2 + 4n + 3$$

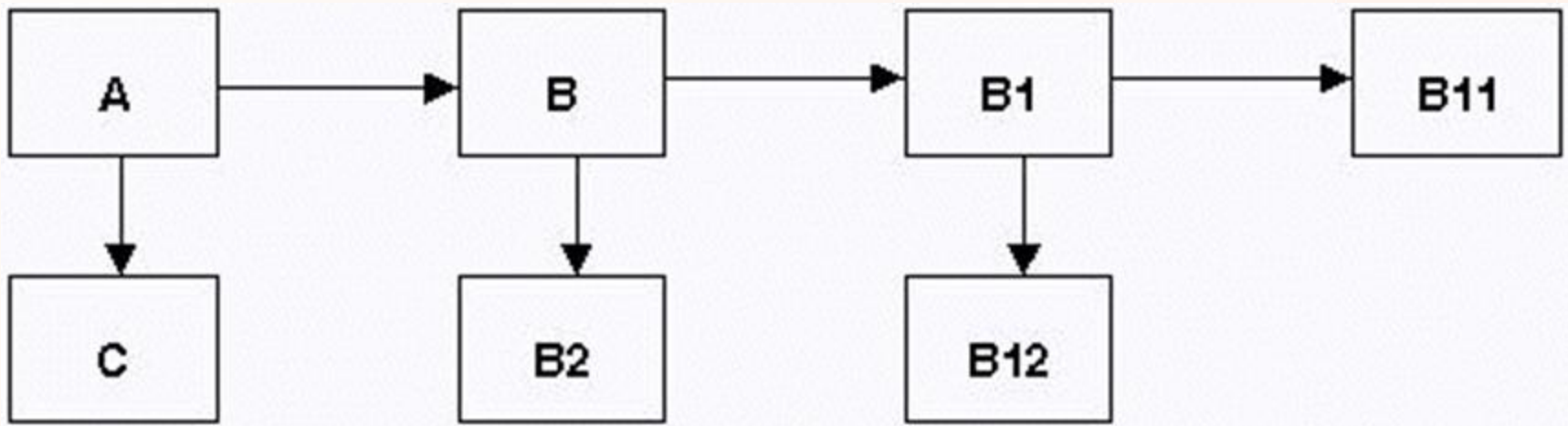


$$T(n) = O(n^2)$$

Đánh giá tính hiệu quả về thời gian



Chương trình có gọi chương trình con không đệ quy



Sơ đồ gọi thực hiện các chương trình con không đệ quy

Đánh giá tính hiệu quả về thời gian

