

CÁC TÍNH CHẤT CỦA THUẬT TOÁN



CÁC TÍNH CHẤT CỦA THUẬT TOÁN (TT)

1

ĐẦU VÀO/ ĐẦU RA



2

TÍNH HIỆU QUẢ

3

TÍNH TỔNG QUÁT



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

PHƯƠNG PHÁP BIỂU DIỄN THUẬT TOÁN

01

NGÔN NGỮ TỰ NHIÊN

02

LƯU ĐỒ KHỐI



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

03

MÃ GIẢ (PSEUDO-CODE)

BIỂU DIỄN THUẬT TOÁN - NGÔN NGỮ TỰ NHIÊN

Xét ví dụ: Giải phương trình bậc 1

Nhập: a, b ;

Xuất: kết quả nghiệm ;

Bắt đầu

Bước 1: Nhập a, b

Bước 2: Nếu $a = 0$

- 2.1 Nếu $b = 0$ thì phương trình vô định. Kết thúc thuật toán;
- 2.2 Nếu $b \neq 0$ thì phương trình vô nghiệm. Kết thúc thuật toán;

Bước 3: Nếu $a \neq 0$ thì thực hiện phép tính $x = -b/a$

Bước 4: Xuất kết quả và kết thúc thuật toán

Kết thúc.



BIỂU DIỄN THUẬT TOÁN - LƯU ĐỒ KHỐI

5.2 Lưu đồ khối

Một số quy ước:

Các thao tác nhập/ xuất dữ liệu

Ví dụ: Nhập a, b,...

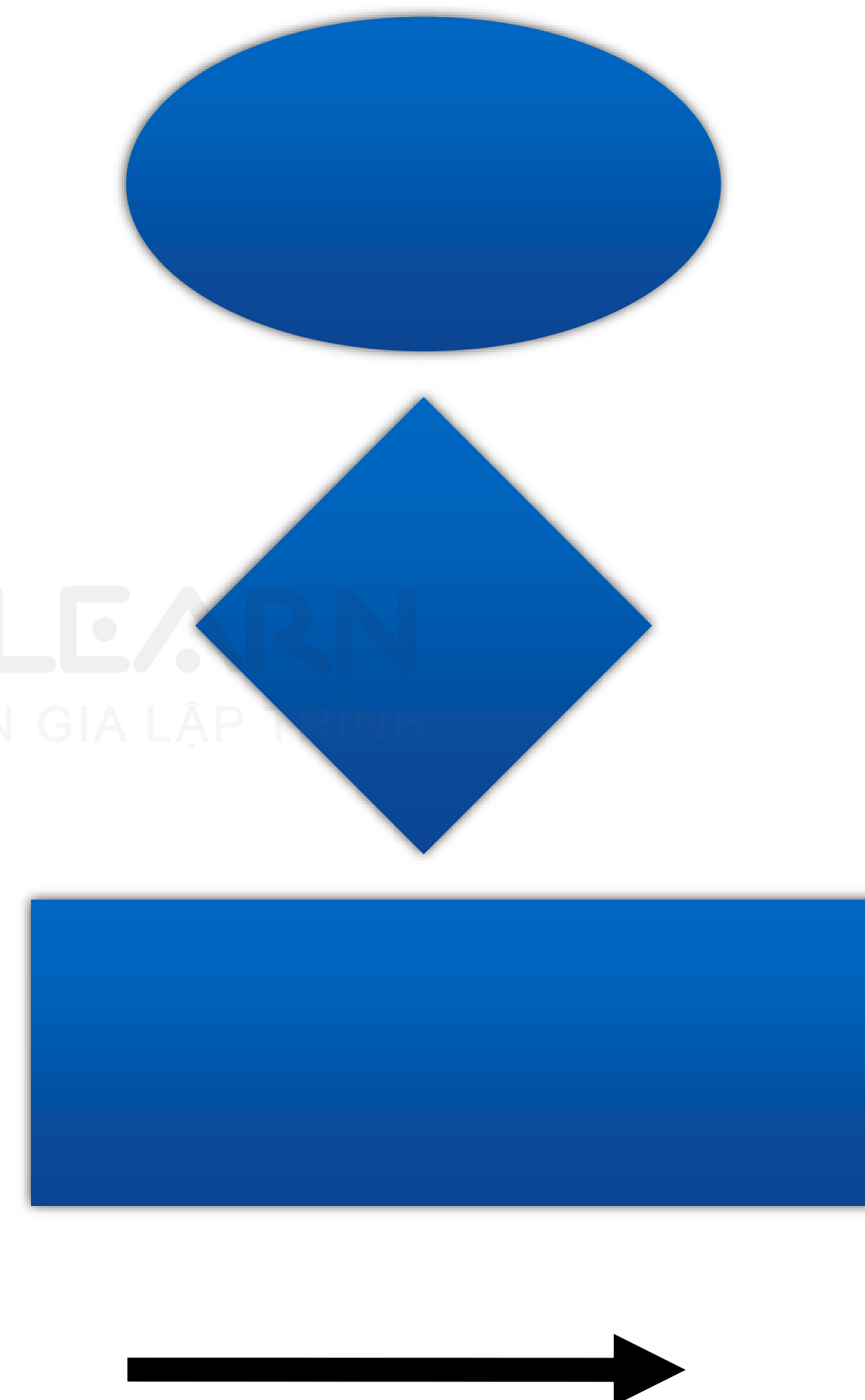
Các Lựa chọn/ Điều kiện/ So sánh

Ví dụ: So sánh a và b có $= 0$

Các Thao tác xử lý/ phép toán

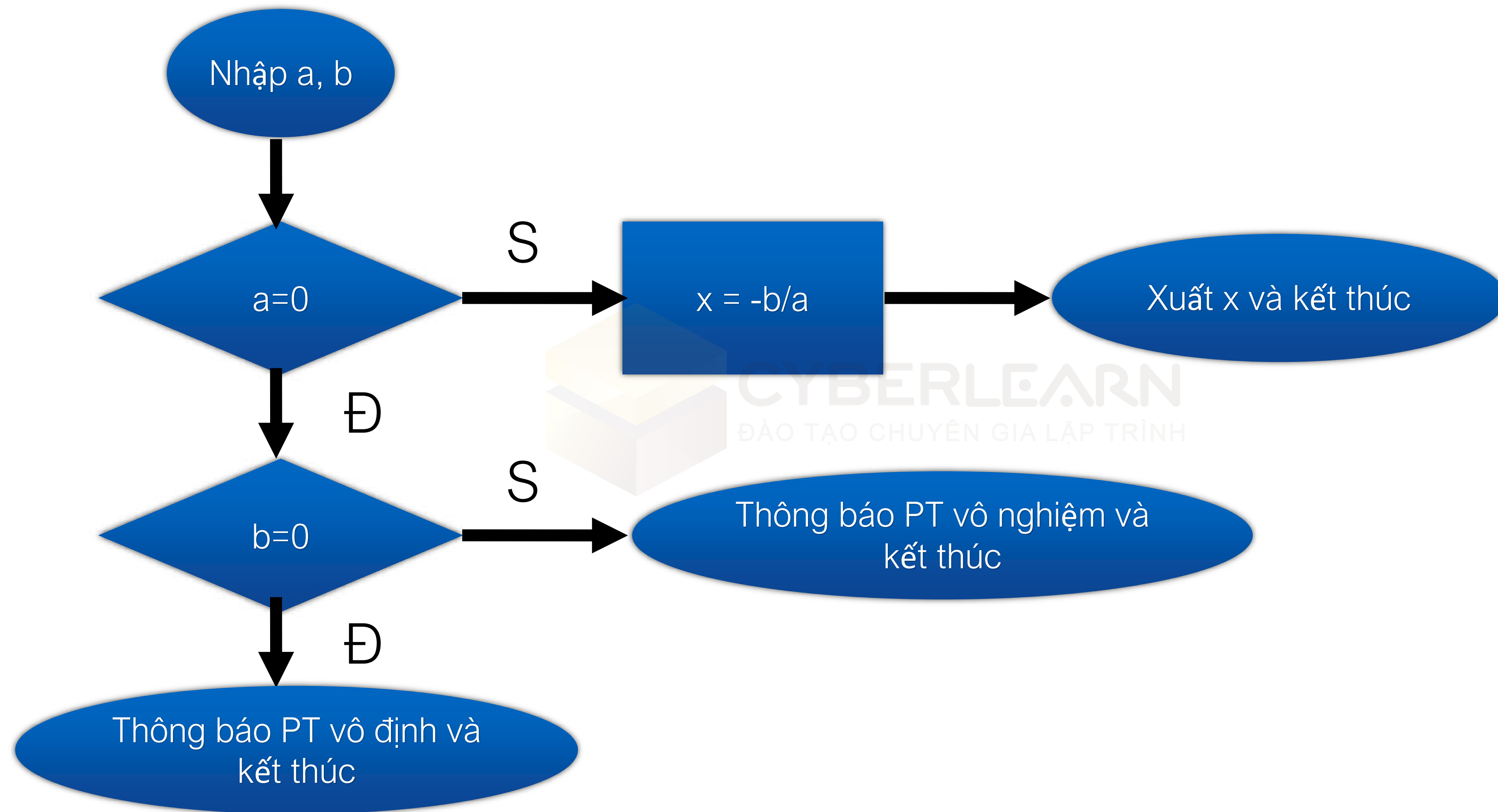
Ví dụ: Tính $x = -b/a$, tăng i lên 1

Các đường đi/ qui trình thực hiện



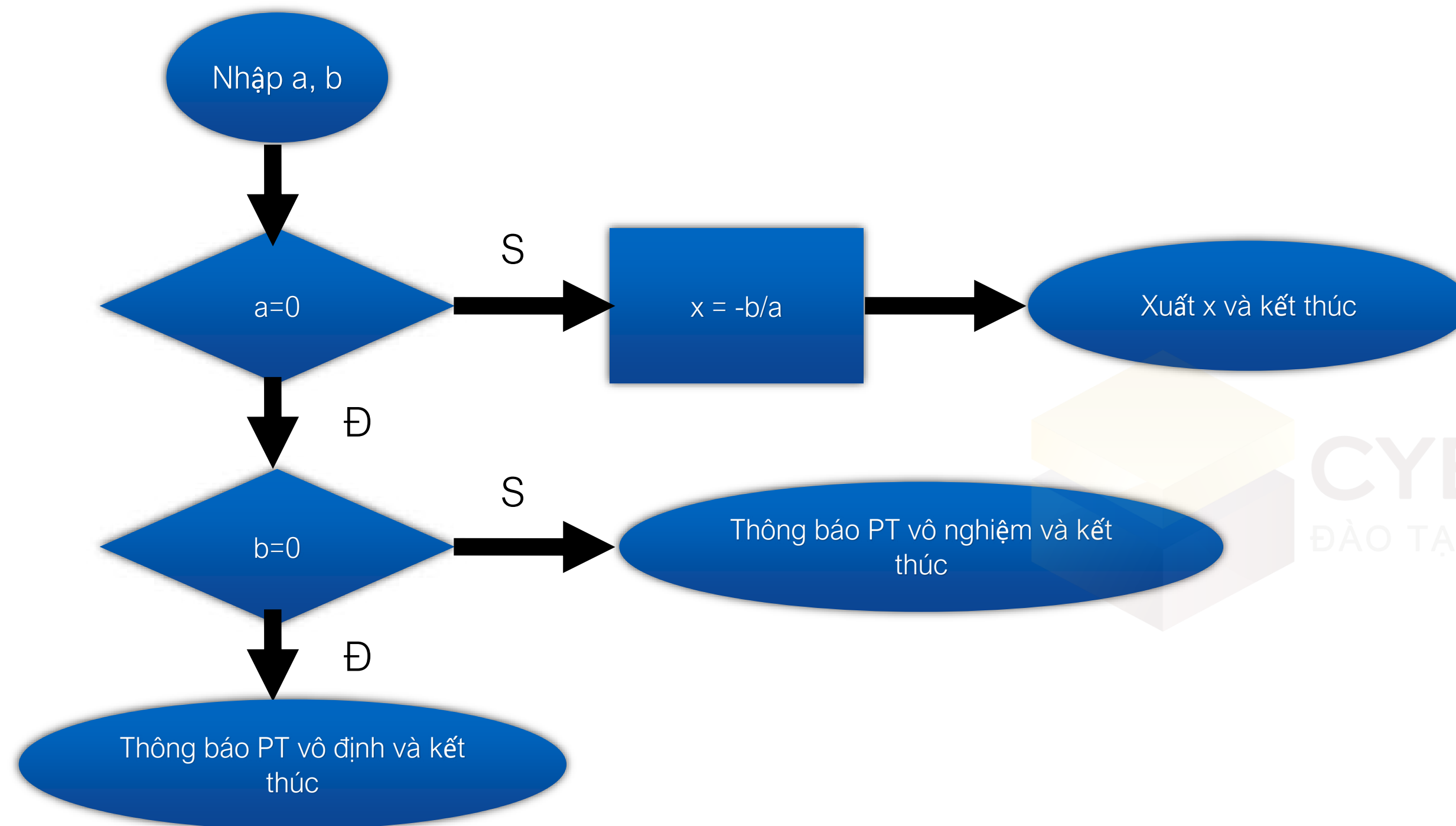
BIỂU DIỄN THUẬT TOÁN - LƯU ĐỒ KHỐI

Xét ví dụ: Giải phương trình bậc 1



BIỂU DIỄN THUẬT TOÁN - LƯU ĐỒ KHỐI

So sánh 2 phương pháp



Nhập: a, b ;

Xuất: kết quả nghiệm ;

Bắt đầu

Bước 1: Nhập a, b

Bước 2: Nếu a = 0

- 2.1 Nếu b = 0 thì phương trình vô định. Kết thúc thuật toán;

- 2.2 Nếu b ≠ 0 thì phương trình vô nghiệm.

Kết thúc thuật toán;

Bước 3: Nếu a ≠ 0 thì thực hiện phép tính $x = -b/a$

Bước 4: Xuất kết quả và kết thúc thuật toán

Kết thúc.

BIỂU DIỄN THUẬT TOÁN

MÃ GIẢ (PSEUDO-CODE)

Sử dụng các từ khóa vay mượn các ngôn ngữ lập trình: Như if, else, for, while,...

Input: a,b ;

Output: kết quả nghiệm ;

Begin

read a, b

if a = 0

if b = 0

write “PT vô định”

else

write “PT vô nghiệm”

else

x= -b/a

write x

End



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

ĐỘ PHỨC TẠP THUẬT TOÁN

Algorithm



SỐ PHÉP TOÁN THUẬT TOÁN THỰC THI

```
int[] arr = {1,3, 4, 9, 8};
```

```
n = arr.length;
```

```
for ( i = 0; i < n; i++ ) {  
    System.out.println( arr[i] );  
}
```

1 phép tính

1 phép tính

Chạy n step

Mỗi step

i < n : 1 phép tính

In arr[i] : 1 phép tính

i++ : 1 phép tính

Mối quan hệ giữa input đầu vào và số lượng phép tính là hàm:

$$T(n) = 3n + 2 \Rightarrow \text{Độ phức tạp của thuật toán}$$



SỐ PHÉP TÍNH THUẬT TOÁN THỰC THI (TT)

```
i = 1;  
n = 1000;
```

```
while ( i < n ) {  
    System.out.println( i );  
    i *= 2;  
}
```

2 phép tính

Chạy n step

Mỗi step

i < n : 1 phép tính

print (i) : 1 phép tính

i *= 2 : 1 phép tính

$3\log_2(n)$

i *= 2 => 1 -> 2 -> 4 -> 8 ->

$2^t = n \Rightarrow t = \log_2(n)$

Mối quan hệ giữa input đầu vào và số lượng phép tính là hàm:

$$T(n) = 3\log_2(n) + 2 \Rightarrow \text{Độ phức tạp của thuật toán}$$



SỐ PHÉP TOÁN THUẬT TOÁN TRONG THỰC TẾ

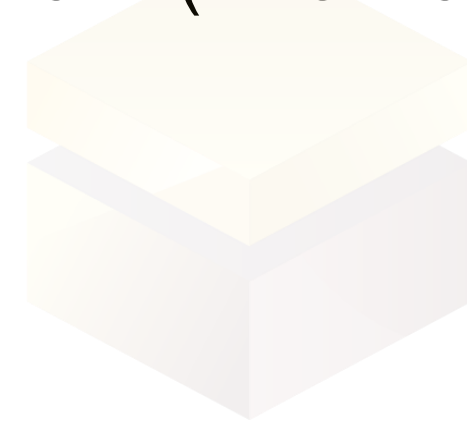
$$T(n) = n^4 + 28 \cdot n^3 + 22 \cdot n^2 + 50 \cdot n + 298 \cdot \log_2(3n) + 12000$$



PHÂN TÍCH ĐỘ PHỨC TẠP

Big O Notation (Kí hiệu O) - hay còn gọi là ký hiệu Landau. Ký hiệu lấy từ tên nhà toán học Landau.

Big O nghĩa là tốc độ tăng nhanh của hàm (The Rate of Growth of function)



ĐÀO TẠO CHUYÊN GIA



Edmund
Landau

1877 - 1938



ĐƠN GIẢN HÓA SỰ PHỨC TẠP

$$T(n) = \cancel{2} + \cancel{3}n \longrightarrow T(n) = O(n)$$

$$T(n) = \cancel{4}n^2 + \cancel{2}n + \cancel{5} \longrightarrow T(n) = O(n^2)$$

$$T(n) = \cancel{3}\log_2(n) + \cancel{2} \longrightarrow T(n) = O(\log_2(n))$$

Big O loại bỏ các biến số đầu vào có ảnh hưởng nhỏ và giữ lại các biến số có ảnh hưởng lớn khi số lượng đầu vào tiến tới vô cùng

QUI TẮC TÌM BIG-O

Quy tắc hằng

$$O(c * f(n)) = O(f(n))$$

Quy tắc cộng

$$O(f(n) + g(n)) = O(\max(f(n), g(n)))$$

Quy tắc nhân

$$O(f(n) * g(n)) = O(f(n)) * O(g(n))$$

$$T(n) = 3n + 2 \Rightarrow O(2 + 3n) = O(3n) = O(n)$$

$$T(n) = 4n^2 + 2n + 5 \Rightarrow O(4n^2 + 2n + 5) = O(4n^2) = O(n^2)$$

$$T(n) = 3\log_2(n) + 2 \Rightarrow O(3\log_2(n)) = O(\log_2(n))$$



ĐỘ PHỨC TẠP THUẬT TOÁN

Độ phức tạp	Thuật ngữ/tên phân lớp
$O(1)$	Độ phức tạp hằng số
$O(\log_2 n)$	Độ phức tạp logarit
$O(n)$	Độ phức tạp tuyến tính
$O(n \log_2 n)$	Độ phức tạp $n \log_2 n$
$O(n^a)$	Độ phức tạp đa thức
$O(a^n), a > 1$	Độ phức tạp hàm mũ
$O(n!)$	Độ phức tạp giai thừa

ĐỘ PHỨC TẠP THUẬT TOÁN

	logn	n	nlogn	n^2	2^n	n!
10	$3 \cdot 10^{-9}$	10^{-8}	$3 \cdot 10^{-8}$	10^{-7}	10^{-6}	$3 \cdot 10^{-3}$
10^2	$7 \cdot 10^{-9}$	10^{-7}	$7 \cdot 10^{-7}$	10^{-5}	$4 \cdot 10^{13}$ năm	*
10^3	$1,0 \cdot 10^{-8}$	10^{-6}	$1 \cdot 10^{-5}$	10^{-3}	*	*
10^4	$1,3 \cdot 10^{-8}$	10^{-5}	$1 \cdot 10^{-4}$	10^{-1}	*	*
10^5	$1,7 \cdot 10^{-8}$	10^{-4}	$2 \cdot 10^{-3}$	10	*	*
10^6	$2 \cdot 10^{-8}$	10^{-3}	$2 \cdot 10^{-2}$	17 phút	*	*

- Lưu ý:
 - Mỗi phép toán giả sử thực hiện trong 10^{-9} giây (~ CPU 1GHz).
 - *: thời gian lớn hơn 100^{100} năm