

# Kiểm tra tính đúng đắn của chương trình bằng bộ test

CS112.L12.KHCL - Nhóm 9



# Nội dung chính

1. Chương trình là gì?
2. Tính đúng đắn của chương trình là gì?
3. Cách kiểm tra tính đúng đắn
4. Cách tạo bộ test



Chương trình là gì?



# Chương trình

Là một công cụ chạy trên máy tính, được viết bằng ngôn ngữ lập trình (C++, Python...) và sử dụng một (hoặc nhiều) thuật toán để giúp người dùng giải viết một vấn đề nào đó



Tính đúng đắn của chương  
trình là gì?



# Khi nào thì chương trình đúng

Các lỗi thường gặp trong chương trình

- Lỗi cú pháp (đã có IDE kiểm tra)
- Lỗi thực thi
  - Chia 0
  - Lặp vô tận
  - Truy cập ngoài mảng
  - ...

# Khi nào thì chương trình đúng

Thuật toán đúng là khi nó cho ra **kết quả mong đợi** với mọi dữ liệu đầu vào **hợp lệ** trong một khoảng **thời gian có giới hạn**

# Kết quả mong đợi

Tính số Fibonacci thứ N.

Lấy kết quả tìm được chia  $10^9 + 7$  và in ra màn hình phần dư

N = 10	→ 55	→ 55
N = 69	→ 117669030460994	→ 29637311



# Dữ liệu đầu vào hợp lệ

- Là những tham số nằm trong phạm vi yêu cầu bài toán

Chương trình (thuật toán) đúng là khi nó cho ra **kết quả mong đợi với mọi dữ liệu đầu vào hợp lệ** trong một khoảng **thời gian có giới hạn**

# Dữ liệu đầu vào hợp lệ

**Xét bài toán sau:** Tìm K trong mảng A gồm N phần tử

**Yêu cầu:**

- $1 \leq N \leq 1\,000$
- $A[i]$  là kiểu unsigned int 32 bit
- K cùng kiểu dữ liệu với  $A[i]$

# Dữ liệu đầu vào hợp lệ

**Xét bài toán sau:** Tìm K trong mảng A gồm N phần tử

**Hợp lệ:**

$$N = 200$$

$$A = [4, 2, 9]$$

$$K = 20$$

**Không hợp lệ:**

$$N = 1300$$

$$A = [-1, -2, "a"]$$

$$K = 20.4$$

# Dữ liệu đầu vào hợp lệ

**Xét bài toán sau:** Tìm K trong mảng A gồm N phần tử

**Hợp lệ:**

$N = 200$

$A = [4, 2, 9]$

$K = 20$

**Không hợp lệ:**

$N = 1300$

$A = [-1, -2, "a"]$

$K = 20.4$

Nếu nhập vào input không hợp lệ thì output đầu ra của chương trình là gì cũng không không ảnh hưởng đến tính đúng đắn của chương trình

# Khoảng thời gian có giới hạn

- Chương trình phải trả về kết quả sau một khoảng thời gian nào đó.

Chương trình (thuật toán) đúng là khi nó cho ra **kết quả mong đợi với mọi dữ liệu đầu vào hợp lệ** trong một khoảng **thời gian có giới hạn**

# Khoảng thời gian có giới hạn

## Ví dụ:

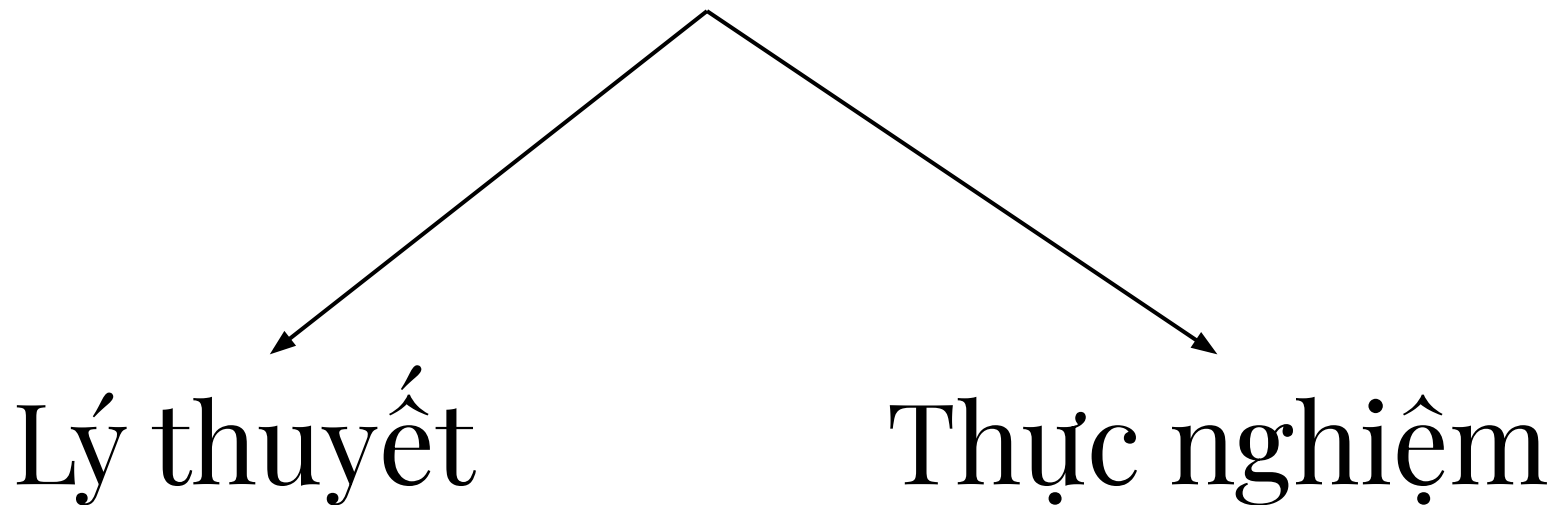
- Chương trình cộng 2 số nguyên sẽ cho ra kết quả trong 5 giây là đúng
- Chương trình dò password nếu cho ra kết quả trong 100 năm thì vẫn đúng



# Cách kiểm tra tính đúng đắn



# Làm sao để kiểm tra tính đúng đắn





# Lý thuyết

**Bài tập 1:** Chứng minh rằng với  $n \in \mathbb{N}^*$  thì  $1 + 3 + 5 + \dots + (2n - 1) = n^2$  (1)

*Bài giải:*

Kiểm tra khi  $n = 1$ : mệnh đề (1) trở thành:  $1 = 1^2 = 1$  (đúng)

Giả sử mệnh đề (1) đúng khi  $n = k \geq 1$ , tức là:

$$S_k = 1 + 3 + 5 + \dots + (2k - 1) = k^2 \text{ (giả thiết quy nạp)}$$

Cần chứng minh mệnh đề (1) đúng với  $n = k + 1$ , tức là cần chứng minh:

$$S_{k+1} = 1 + 3 + 5 + \dots + (2k - 1) + 2[2(k + 1) - 1] = (k + 1)^2$$

$$\text{Thật vậy: } S_{k+1} = S_k + [2(k + 1) - 1] = k^2 + 2k + 1 = (k + 1)^2$$

Vậy mệnh đề (1) đúng với mọi  $n \in \mathbb{N}^*$ .

# Thực nghiệm

Viết chương trình tính tổng:

$$S(n) = 1 + 3 + 5 + \dots + (2n - 1)$$

- $1 \leq n \leq 1\,000\,000$

Input	Output
1	1
2	4
...	...
1 000 000	1 000 000 000 000 000

Chương trình (thuật toán) đúng là khi nó cho ra **kết quả mong đợi với mọi dữ liệu đầu vào hợp lệ** trong một khoảng **thời gian có giới hạn**

# Thực nghiệm

Viết chương trình tính tổng:

$$S_n = 1 + 3 + 5 + \dots + (2n - 1)$$

- $1 \leq n < \infty$

# Làm sao để kiểm tra tính đúng đắn

	Lý thuyết	Thực tế
Ưu	Đảm bảo thuật toán đúng 100%	Dễ thực hiện
Nhược	Khó thực hiện	Không đảm bảo thuật toán đúng 100% nếu không xét hết tất cả các trường hợp

# Cách tạo bộ test

# Ví dụ 1

Vừa gà vừa chó  
Bó lại cho tròn  
X con  
Y chân chẵn

# Đề bài

## Input:

- X: Số gà + số chó  $2 \leq X \leq 25 \times 10^7$
- Y: Số chân gà + số chân chó  $4 \leq Y \leq 10^9$

## Output:

- Số gà  $\geq 0$
- Số chó  $\geq 0$

INPUT	OUTPUT
36	22
100	14

## Yêu cầu:

- Không tìm thấy kết quả thỏa yêu cầu đề bài thì trả về -1



# Ví dụ 1

1. Trường hợp không tầm thường - biên nhỏ nhất

# Ví dụ 1

## 1. Trường hợp không tầm thường - biên nhỏ nhất

INPUT	OUTPUT
2	2
4	0

- X: Số gà + số chó  $2 \leq X \leq 25 \cdot 10^7$
- Y: Số chân gà + số chân chó  $4 \leq Y \leq 10^9$

# Ví dụ 1

## 1. Trường hợp không tầm thường - biên lớn nhất

INPUT	OUTPUT
250_000_000 1_000_000_000	0 250_000_000

- X: Số gà + số chó  $2 \leq X \leq 25 \cdot 10^7$
- Y: Số chân gà + số chân chó  $4 \leq Y \leq 10^9$

# Ví dụ 1

## 2. Trường hợp tầm thường - khoảng giữa

INPUT	OUTPUT
61 216	14 47
51 174	15 36
64 230	13 51
...	...

# Đề bài

## Input:

- X: Số gà + số chó  $2 \leq X \leq 25 \times 10^7$
- Y: Số chân gà + số chân chó  $4 \leq Y \leq 10^9$

## Output:

- Số gà
- Số chó

INPUT	OUTPUT
36	22
100	14

## Yêu cầu:

- Không tìm thấy kết quả thỏa yêu cầu đề bài thì trả về -1

# Ví dụ 1

## 2 gợi ý

- Trường hợp không tầm thường (lớn, bé)
- Trường hợp tầm thường (khoảng giữa)

# Ví dụ 2

Viết chương trình nhân 2 ma trận

# Đề bài

## **Input:**

- $m, n, p, q$ : số dòng, cột ma trận A và B ( $1 \leq m, n, p, q \leq 4000$ )
- A: mảng số nguyên 2 chiều ( $m \times n$ )
- B: mảng số nguyên 2 chiều ( $p \times q$ )

## **Output:**

- Ma trận kết quả của  $A * B$

## **Yêu cầu:**

- Khi hai ma không nhân được với nhau thì trả về -1



# Ví dụ 2

## 1. Trường hợp không tầm thường - biên nhỏ nhất

INPUT	OUTPUT
<div>1 1</div> <div>2</div> <div>1 1</div> <div>2</div>	<div>4</div>

$m, n, p, q$ : số dòng, cột ma trận 1 và 2 ( $1 \leq m, n, p, q \leq 4000$ )

# Ví dụ 2

## 2. Trường hợp không tầm thường - biên lớn nhất

INPUT	OUTPUT																											
<div>4000 4000      4000 4000</div> <div><table><tr><td>36</td><td>...</td><td>41</td></tr><tr><td>...</td><td>...</td><td>...</td></tr><tr><td>42</td><td>...</td><td>13</td></tr></table><table><tr><td>1</td><td>...</td><td>2</td></tr><tr><td>...</td><td>...</td><td>...</td></tr><tr><td>3</td><td>...</td><td>5</td></tr></table></div>	36	...	41	...	...	...	42	...	13	1	...	2	...	...	...	3	...	5	<div></div> <div><table><tr><td>1</td><td>...</td><td>2</td></tr><tr><td>...</td><td>...</td><td>...</td></tr><tr><td>3</td><td>...</td><td>5</td></tr></table></div>	1	...	2	...	...	...	3	...	5
36	...	41																										
...	...	...																										
42	...	13																										
1	...	2																										
...	...	...																										
3	...	5																										
1	...	2																										
...	...	...																										
3	...	5																										

$m, n, p, q$ : số dòng, cột ma trận 1 và 2 ( $1 \leq m, n, p, q \leq 4000$ )

# Ví dụ 2

## 3. Trường hợp tầm thường - khoảng giữa (nên tạo test có dữ liệu không đối xứng)

`m = random()`

`n = random()`

`p = n`

`q = random()`

INPUT						OUTPUT		
m n p q								
36	...	41	1	...	2	1	...	2
...	...	...	...	...	...	...	...	...
42	...	13	3	...	5	3	...	5

`m, n, p, q`: số dòng, cột ma trận 1 và 2 ( $1 \leq m, n, p, q \leq 4000$ )

# Ví dụ 2

## 3. Các trường hợp đặc biệt của bài toán

$$OA = AO = 0$$

$$IA = AI = A$$

# Ví dụ 2

3.  $IA = AI = A$

$$\begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array} * \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}$$

$$\begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array} * \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}$$

# Ví dụ 2

3.  $A0 = A0 = 0$

$$\begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array} * \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}$$

$$\begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array} * \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}$$

# Ví dụ 3

Sắp xếp mảng tăng dần

# Đề bài

## **Input:**

- N: số phần tử trong mảng ( $N \leq 1\,000\,000$ )
- Arr[N]: chứa các phần tử là các số nguyên dương

## **Output:**

- In ra mảng gồm các phần tử xếp tăng dần



# Ví dụ 3

## 1. Test biên nhỏ nhất, lớn nhất

INPUT	OUTPUT
1	100
100	

INPUT	OUTPUT
1000000000	3 4 5 ... 10
5 3 4 ... 10	

# Ví dụ 3

## 2. Test khoảng giữa

INPUT	OUTPUT
1000 2 98 4 24 0 120 ... 453 234	0 2 4 24 98 120 ... 234 453
1000000 3 123 34 ... 23445 120345 2	2 3 34 123 ... 23445 120345
124578 9999 123 4345 28 ... 45 25 25	25 25 28 45 123 ... 4345 9999

# Ví dụ 3

## 3. Test hiệu năng - đặt thời gian chạy tối đa cho chương trình

- Chia các bộ test làm 4 phần (có thể chia thành nhiều phần hơn)
  - Phần 1: Gồm các bộ test với  $N \approx 250\ 000$
  - Phần 2: Gồm các bộ test với  $N \approx 500\ 000$
  - Phần 3: Gồm các bộ test với  $N \approx 750\ 000$
  - Phần 4: Gồm các bộ test với  $N \approx 1\ 000\ 000$

# Ví dụ 3

## *Nhận xét:*

- Chỉ pass được phần 1 → chưa tối ưu :(
- Pass được phần 1, 2, 3 → tối ưu vừa vừa :|
- Pass tất cả các phần → tối ưu quá tốt ;)

Giải đáp

# Tổng kết

1. Khi nào thì chương trình đúng
2. Cách chứng minh chương trình đúng: lý thuyết, thực nghiệm
3. 3 gợi ý để tạo bộ test

Cảm ơn các bạn lắng nghe