

# THỰC HÀNH

## MÔN GIẢI THUẬT XỬ LÝ SONG SONG VÀ PHÂN BỐ

### I. MỤC TIÊU

- Hiểu và vận dụng các khái niệm cơ bản về lập trình song song để giải quyết một bài toán tính toán có độ phức tạp cao.
- Nắm vững quy trình thiết kế, hiện thực hóa và đánh giá hiệu năng của một ứng dụng song song.

### II. YÊU CẦU BÀI TOÁN

#### 1. MẢNG TỔNG TIỀN TỐ

**Challenge 0:** Nhận vào 1 mảng số nguyên A size N ( $N \leq 10^6$ ), hãy tính mảng tổng tiền tố của mảng A này.

Ví dụ:

$$A = [1, 2, 3, 4]$$

Return [1, 3, 6, 10]

#### 2. SỐ FIBO

**Challenge 1:**

Ta biết công thức số fibo:

$$\begin{aligned}F(0) &= F(1) = 1, \\F(i) &= (F(i - 1) + F(i - 2)) \bmod Q, 2 \leq i\end{aligned}$$

Nhiệm vụ của bạn là xác định  $F(A)$ , với số A cho trước.

**Input:**

Dòng đầu tiên gồm 2 số nguyên dương  $N, Q, 1 \leq N \leq 10^6$ , lần lượt là tổng số truy vấn và số moduler Q.

N dòng tiếp theo, dòng thứ i chứa số nguyên dương  $A_i, 2 \leq A_i \leq 2 \times 10^8$ , là thứ tự số fibo cần tìm.

**Output:**

Function `MAIN(input_file_path)` trả về một chuỗi gồm N số nguyên, với phần tử thứ i là số  $F(A_i)$

**Ví dụ:**

**Input:**

3 1000000000

3

4

5

**Output:**

3

5

8

### **3. NHÂN 2 MA TRẬN**

**Challenge 2:** Cho hai ma trận vuông A và B có cùng kích thước  $n \times n$  ( $n \leq 10000$ ).

Mục tiêu là thiết kế và hiện thực hóa một chương trình song song để tính tích của hai ma trận này, ký hiệu là  $C = A \times B$

**Input:** các phần tử của hai ma trận A và B làm đầu vào.

**Output:** ma trận kết quả C.

Ví dụ:

**Input:**

Ma trận A: 1 2 3 4

Ma trận B: 5 6 7 8

**Output**

Ma trận C: 19 22 43 50

### **4. SẮP XẾP MẢNG MỘT CHIỀU**

**Challenge 3:** Cho một mảng A gồm n phần tử số nguyên ( $n \leq 10^6$ ). Mục tiêu là thiết kế và hiện thực hóa một chương trình song song để sắp xếp các phần tử của mảng A theo thứ tự tăng dần.

**Input:** Mảng số nguyên A size n

**Ouput:** Mảng A đã được sắp xếp theo thứ tự tăng dần

Ví dụ:

$A = [5, 2, 9, 4]$

Return [2, 4, 5, 9]

### **5. TÌM KIẾM TRONG MẢNG SỐ NGUYÊN LỚN (NHẬP TỪ FILE)**

**Challenge 4:** Cho một file văn bản chứa một dãy rất lớn các số nguyên, cách nhau bởi dấu cách hoặc dấu xuống dòng. Mục tiêu là đọc các số nguyên này vào một mảng A. Sau đó, cho một giá trị khóa K (là một số nguyên), hãy tìm xem giá trị K có tồn tại trong mảng A hay không bằng cách sử dụng một thuật toán tìm kiếm song song. Nếu có, trả về chỉ số (một trong các chỉ số nếu có nhiều lần xuất hiện) của K trong mảng. Nếu không tồn tại, trả về -1.

**Input:** Tên file chứa mảng số nguyên, Giá trị khóa (K)

- **Tên file chứa mảng số nguyên:** Một chuỗi ký tự đại diện cho tên của file văn bản đầu vào. File này chứa một hoặc nhiều dòng, mỗi dòng chứa một hoặc nhiều số nguyên cách nhau bởi dấu cách hoặc dấu xuống dòng. Ví dụ, file có thể có nội dung như sau:

10 25 5 30 15

40 20

5 35 45 0

... (và tiếp tục với hàng triệu số nguyên)

- **Giá trị khóa (K):** Một số nguyên K là giá trị cần tìm kiếm trong mảng số nguyên đã đọc từ file. Giá trị này có thể được nhập trực tiếp từ người dùng hoặc cũng có thể được đọc từ một file khác.

**Output:**

Chương trình sẽ xuất ra một trong hai kết quả sau:

- **Nếu tìm thấy khóa K:** Chỉ số (vị trí) của **một** lần xuất hiện của K trong mảng A (mảng được tạo sau khi đọc dữ liệu từ file). Các chỉ số thường bắt đầu từ 0.
- **Nếu không tìm thấy khóa K:** Giá trị -1.

**Ví dụ:**

Giả sử chúng ta có một file văn bản tên là data.txt với nội dung sau:

10 25 5 30 15

40 20

5 35 45 0

Và giá trị khóa K được nhập là 20.

- **Input:** Tên file: data.txt, Khóa K: 20
- **Quá trình thực hiện:**

Chương trình đọc file data.txt và tạo ra mảng A như sau: [10, 25, 5, 30, 15, 40, 20, 5, 35, 45, 0].

Thuật toán tìm kiếm song song được thực hiện trên mảng A để tìm giá trị 20.

Giá trị 20 được tìm thấy ở chỉ số 6 (tính từ 0).

- **Output:** 6

#### **Một ví dụ khác:**

Giả sử file data.txt có nội dung tương tự, nhưng giá trị khóa K là 100.

- **Input:** Tên file: data.txt, Khóa K: 100
- **Quá trình thực hiện:**

Chương trình đọc file data.txt và tạo ra mảng A: [10, 25, 5, 30, 15, 40, 20, 5, 35, 45, 0].

Thuật toán tìm kiếm song song được thực hiện để tìm giá trị 100.

Giá trị 100 không tồn tại trong mảng A.

- **Output:** -1

### **6. YÊU CẦU TRONG BÁO CÁO CHO 2 CHALLENGE 0, 1, 2, 3, 4:**

Sinh viên thực hiện các mục sau cho từng challenge

#### **a. Phân tích song song hóa:**

- Xác định các phần công việc có thể thực hiện song song trong thuật toán ở các thử thách.
- Đề xuất một hoặc nhiều phương pháp phân chia công việc cho các tiến trình/luồng xử lý cho từng bài toán
- Mô tả chi tiết cách dữ liệu được phân phối và truy cập bởi các tiến trình/luồng.
- Phân tích các yếu tố có thể ảnh hưởng đến hiệu năng của chương trình song song, chẳng hạn như sự phụ thuộc dữ liệu, cân bằng tải, và chi phí giao tiếp (nếu có).

#### **b. Thiết kế thuật toán song song:**

- Trình bày thuật toán song song chi tiết cho phương pháp phân chia công việc đã chọn.
- Sử dụng sơ đồ hoặc lưu đồ (nếu cần) để minh họa luồng thực hiện của thuật toán.

#### **c. Đánh giá hiệu năng:**

- Thực hiện đo thời gian thực thi của chương trình song song. Với bài toán nhân 2 ma trận thử các kích thước ma trận khác nhau (ví dụ:  $n=128,256,512,1024,5000$ ). Với bài toán sắp xếp thử các kích thước mảng khác nhau (ví dụ:  $n=10^4,10^5,10^6$ ).
- So sánh hiệu năng của chương trình song song với một chương trình tuần tự tương ứng.
- Giải thích các kết quả thu được và thảo luận về các yếu tố ảnh hưởng đến hiệu năng.

### **III. NỘP BÀI VÀ ĐÁNH GIÁ**

#### **1. MÃ NGUỒN**

- Các nhóm nộp mã nguồn Python cho bài toán lập trình song song đã viết (bản cuối cùng) lên hệ thống đã cung cấp, **không** nộp mã nguồn chạy tuần tự lên hệ thống. Sử dụng key đã cấp cho từng nhóm để nộp bài. Mã nguồn cần qua hết các tests. Mã nguồn cần nộp lên hệ thống trước ngày báo cáo.
- **Mã nguồn cần chạy trên server:** Pass all test.
- Nộp mã nguồn theo các challenges tương ứng với ID: [https://challenges.iec-uit.com/challenge\\_<id>](https://challenges.iec-uit.com/challenge_<id>)
- Các bạn xem leaderboard của challenges tương ứng với ID: [https://challenges.iec-uit.com/challenge\\_<id>/leaderboard](https://challenges.iec-uit.com/challenge_<id>/leaderboard)

#### **2. BÁO CÁO VÀ TRÌNH BÀY**

##### **a. Báo cáo**

Trả lời các yêu cầu như đã hướng dẫn, Ghi rõ mã nhóm + key + thông tin các thành viên, Nộp báo cáo theo định dạng **PDF**. Báo cáo sẽ được nộp lên website môn học cùng deadline với nộp mã nguồn.

##### **a. Nội dung trình bày**

Sinh viên tự chuẩn bị slide để trình bày ở buổi báo cáo. Nội dung trình bày gồm:

- Phương pháp phân chia công việc cho các tiến trình/luồng xử lý cho các bài toán trên
- Mô tả chi tiết cách dữ liệu được phân phối và truy cập bởi các tiến trình/luồng theo phương pháp đã chọn

- Trình bày thuật toán song song một cách rõ ràng và chi tiết, dựa trên phương pháp phân chia công việc đã chọn.
- Sử dụng sơ đồ hoặc lưu đồ (nếu cần thiết) để minh họa các bước thực hiện của thuật toán và sự tương tác giữa các tiến trình/luồng.
- Giải thích vai trò của từng bước trong thuật toán và cách chúng đóng góp vào việc tính toán ma trận kết quả và sắp xếp mảng.
- Tiềm năng tăng tốc (speedup) có thể đạt được khi sử dụng thuật toán song song so với thuật toán tuần tự.
- Ví dụ minh họa

#### **b. Đánh giá**

- Mỗi Lab sẽ gồm 1 challenge: 8 điểm
  - o 3 nhóm đứng đầu được toàn bộ điểm của bài
  - o 5 nhóm tiếp theo trừ 1 điểm
  - o 5 nhóm tiếp theo trừ 2 điểm
  - o 5 nhóm tiếp theo trừ 3 điểm.
- Nộp báo cáo report + trình bày ở buổi thực hành tiếp theo: 2 điểm
- Các trường hợp nộp bài trễ, báo cáo trễ, sao chép, gian lận sẽ xử lý tùy mức độ (trừ 10%-100% điểm).