

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ
MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÀI TẬP NHÓM 10
BỞI NHÓM 12

Giáo viên hướng dẫn: *Nguyễn Thanh Sơn*

Nhóm 12:

- *Hoàng Minh Thái - 23521414*
- *Nguyễn Trọng Tất Thành - 23521455*



Mục lục

GIẢI BÀI TẬP BỞI NHÓM 12

Đề Bài 1

Xây dựng thuật toán kiểm tra số nguyên tố song song.

Input: Một số nguyên X .

Output: X có phải là số nguyên tố hay không.

Yêu Cầu

- Kiểm tra kết quả với khi thực hiện tính toán tuần tự.
- So sánh thời gian thực hiện giữa song song và tuần tự trên nhiều test case có cả số nhỏ và lớn.
- Chạy thử và show thời gian thực hiện cả song song và tuần tự trên các test case sau:

- $X = 100000000000000091$
- $X = 100000000000000099$
- $X = 100000000000000049$

Mã Giả Thuật Toán Kiểm Tra Số Nguyên Tố

1.1 Thuật toán Kiểm Tra Số Nguyên Tố Tuần Tự

```
1 def is_prime(X):
2     if X <= 1:
3         return False
4     for i in range(2, int(X ** 0.5) + 1):
5         if X % i == 0:
6             return False
7     return True
```



1.2 Thuật toán Kiểm Tra Số Nguyên Tố Song Song

Sử dụng phương pháp song song để chia công việc cho các tiến trình hoặc luồng.

```
1 import concurrent.futures
2
3 def check_prime_parallel(X):
4     def check_range(start, end):
5         for i in range(start, end):
6             if X % i == 0:
7                 return False
8         return True
9
10    with concurrent.futures.ThreadPoolExecutor() as executor:
11        futures = []
12        # Chia công việc kiểm tra thành các phần vì còn v s đng song song
13        for i in range(2, int(X ** 0.5) + 1, 100):
14            futures.append(executor.submit(check_range, i, min(i + 100, int(X ** 0.5)
15                + 1)))
16
17        # Kiểm tra kết quả song song
18        for future in concurrent.futures.as_completed(futures):
19            if not future.result():
20                return False
21    return True
```

So Sánh Thời Gian Thực Hiện

Chạy thử và đo thời gian thực hiện đối với các test case cho cả thuật toán tuần tự và song song.

```
1 import time
2
3 X = 10000000000000091
4 # Kiểm tra tuần tự
5 start_time = time.time()
6 print(is_prime(X))
7 end_time = time.time()
8 print(f"Thời gian thực hiện tuần tự : {end_time - start_time} giây")
9
10 # Kiểm tra song song
11 start_time = time.time()
12 print(check_prime_parallel(X))
13 end_time = time.time()
14 print(f"Thời gian thực hiện song song: {end_time - start_time} giây")
```

Kết Luận

Thuật toán song song sẽ giúp tăng tốc quá trình kiểm tra số nguyên tố, đặc biệt đối với các số lớn. Tuy nhiên, thời gian thực hiện còn phụ thuộc vào số lượng luồng và tài nguyên hệ thống.



Đề Bài 2

Xây dựng thuật toán nhân ma trận song song.

Input: Hai ma trận A và B .

Output: Kết quả nhân hai ma trận $A \times B$.

Yêu Cầu

- Kiểm tra kết quả với khi thực hiện tính toán tuần tự.
- So sánh thời gian thực hiện giữa song song và tuần tự trên nhiều test case có cả ma trận kích thước nhỏ và lớn.
- Sinh ngẫu nhiên ma trận A, B kích thước 400×400 . Show ra thời gian thực hiện cả song song và tuần tự trên các test case.
- Không sử dụng `numpy.dot()` hay các hàm nhân ma trận code sẵn.

Mã Giải Thuật Toán Nhân Ma Trận

2.1 Thuật toán Nhân Ma Trận Tuần Tự

```
1 import random
2
3 def multiply_matrices(A, B):
4     n = len(A)
5     m = len(B[0])
6     p = len(B)
7
8     # Khi to ma trn kt qu vi gi tr 0
9     result = [[0] * m for _ in range(n)]
10
11     for i in range(n):
12         for j in range(m):
13             for k in range(p):
14                 result[i][j] += A[i][k] * B[k][j]
15     return result
```

2.2 Thuật toán Nhân Ma Trận Song Song

Sử dụng thư viện `concurrent.futures` để chia công việc cho các tiến trình hoặc luồng.

```
1 import concurrent.futures
2
3 def multiply_matrices_parallel(A, B):
4     n = len(A)
5     m = len(B[0])
6     p = len(B)
7
8     result = [[0] * m for _ in range(n)]
9
```



```
10 def calculate_cell(i, j):
11     for k in range(p):
12         result[i][j] += A[i][k] * B[k][j]
13
14     with concurrent.futures.ThreadPoolExecutor() as executor:
15         futures = []
16         for i in range(n):
17             for j in range(m):
18                 futures.append(executor.submit(calculate_cell, i, j))
19
20     # i cc tin trnh song song hon thnh
21     for future in concurrent.futures.as_completed(futures):
22         future.result()
23
24     return result
```

So Sánh Thời Gian Thực Hiện

Chạy thử và đo thời gian thực hiện đối với các test case cho cả thuật toán tuần tự và song song.

```
1 import time
2
3 # Sinh ngu nhin ma trn A, B
4 def generate_matrix(size):
5     return [[random.randint(1, 100) for _ in range(size)] for _ in range(size)]
6
7 A = generate_matrix(400)
8 B = generate_matrix(400)
9
10 # Nhn ma trn tun t
11 start_time = time.time()
12 result = multiply_matrices(A, B)
13 end_time = time.time()
14 print(f"Thi gian thc hin tun t : {end_time - start_time} giy")
15
16 # Nhn ma trn song song
17 start_time = time.time()
18 result_parallel = multiply_matrices_parallel(A, B)
19 end_time = time.time()
20 print(f"Thi gian thc hin song song: {end_time - start_time} giy")
```

Kết Luận

Thuật toán nhân ma trận song song giúp giảm thời gian thực hiện cho các ma trận lớn, đặc biệt là khi có nhiều tiến trình có thể thực hiện song song. Tuy nhiên, thời gian thực hiện còn phụ thuộc vào kích thước ma trận và tài nguyên hệ thống.