1. Bài toán 1: Tính số Fibonacci
2. Mô tả bài toán

Số Fibonacci F(n) là số thỏa mãn:

1. Giải thuật

Sử dụng công thức để tính số Fibonacci thứ n.

Chứng minh công thức:

Đặt số

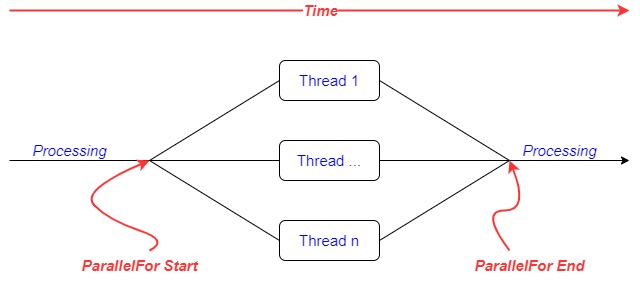
Ta sẽ chứng minh (công thức Binet):

Từ hệ thức truy hồi F(n) - F(n – 1) + F(n – 2) = 0 sẽ dẫn tới phương trình xác định tỉ lệ vàng (là phương trình đa thức đặc trưng của hồi quy):

Có 2 nghiệm phân biệt: .

Khi đó, dãy số{} là nghiệm của công thức đệ quy:

= =



1. Nhận xét

Sử dụng đa luồng tốn thời gian hơn so với đơn luồng.

1. Bài toán 2: Nhân ma trận với ma trận
2. Mô tả bài toán

Nhân hai ma trận A và B lần lượt có kích thước mxa và axn tạo thành ma trận kết quả mxn

1. Giải thuật
2. Giải thuật đơn luồng

Tính toán lần lượt theo công thức ma trận nhân ma trận.

1. Giải thuật đa luồng

Chia việc tính toán giá trị các hàng ở ma trận kết quả cho các luồng:

Diagram

Description automatically generated

1. Nhận xét

Sử dụng đa luồng giúp cho việc tính toán nhanh hơn. Tuy vậy, nếu giảm số cột của ma trận B, cụ thể là dưới 20 (hoặc có thể lớn hơn) thì sử dụng đơn luồng sẽ tốt về mặt thời gian hơn.

1. Bài toán 3: Biến đổi dãy số đã cho trở thành dãy số chỉ có số nguyên tố
2. Mô tả bài toán

Đưa vào một mảng array có n số hạng. Trả về một mảng lastReturn sao cho những số nguyên tố ở mảng array giữ nguyên và những số nguyên tố gần nhất với các hợp số ở mảng array.

1. Giải thuật
2. Giải thuật đơn luồng

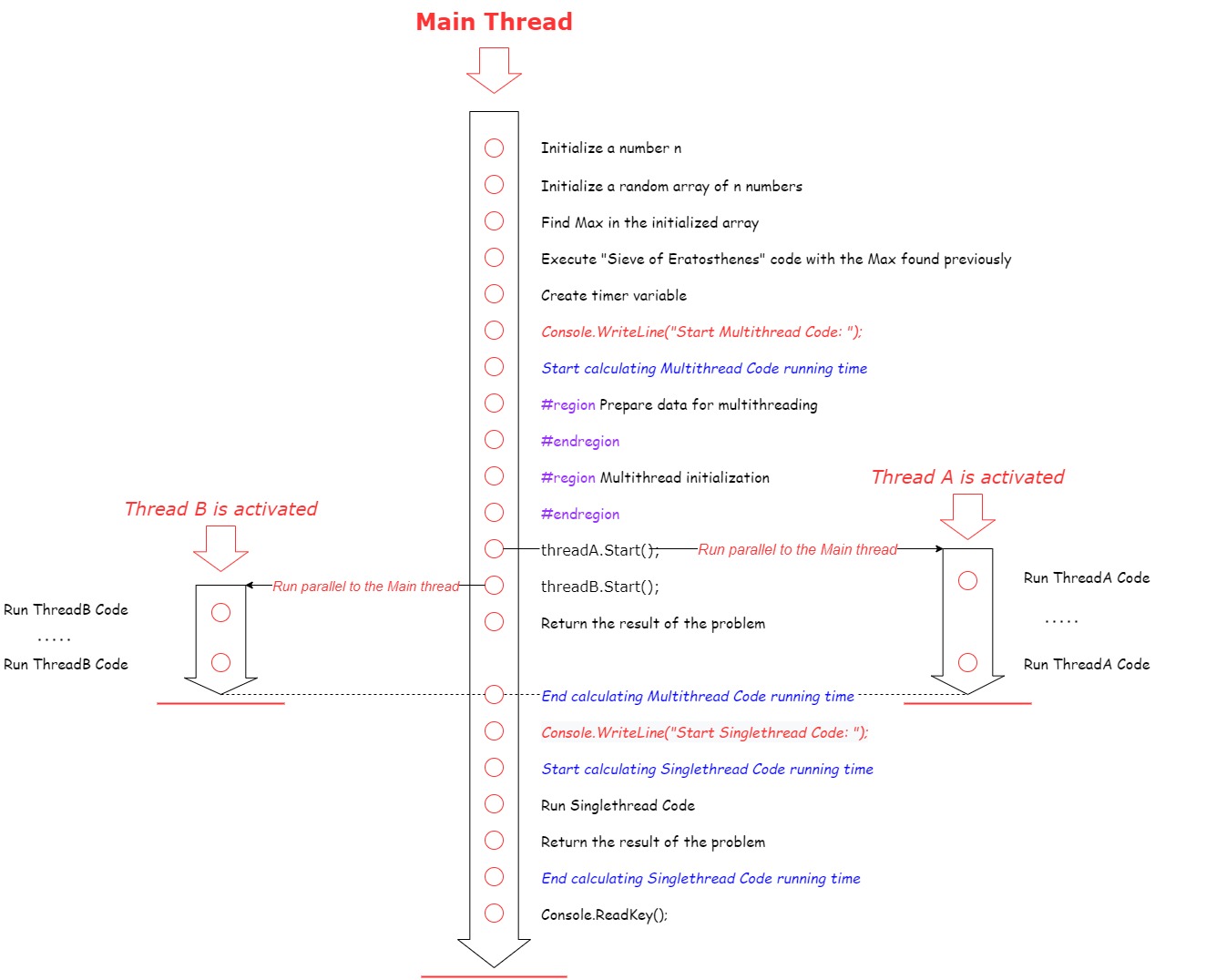
Xét lần lượt từng số trong mảng array, là số nguyên tố thì giữ nguyên, là hợp số thì tìm số nguyên tố gần nhất với số đó và thay thế.

1. Giải thuật đa luồng

Xét tính chẵn lẻ của các số trong mảng array và chia vào 2 mảng con forThreadA chỉ chứa số lẻ và forThreadB chỉ chứa số chẵn.

Khởi tạo hai luồng:

* Thread A sử dụng mảng forThreadA để kiểm tra tính nguyên tố của số, nếu không phải số nguyên tố thì tìm số nguyên tố gần nhất với nó và thay thế.
* Thread B sử dụng mảng forThreadB để trực tiếp tìm số nguyên tố gần nhất với hợp số đang xét và thay thế nó.



1. Nhận xét

* Với kích thước mảng array lớn thì việc sử dụng đa luồng là tối ưu hơn so với đơn luồng (kích thước của mảng lớn hơn 1000)
* Ta có thể tối ưu hơn bằng cách sử dụng nhiều luồng hơn, tuy nhiên, việc chia luồng với kích thước mảng nhỏ sẽ gây mất thời gian hơn so với đơn luồng.