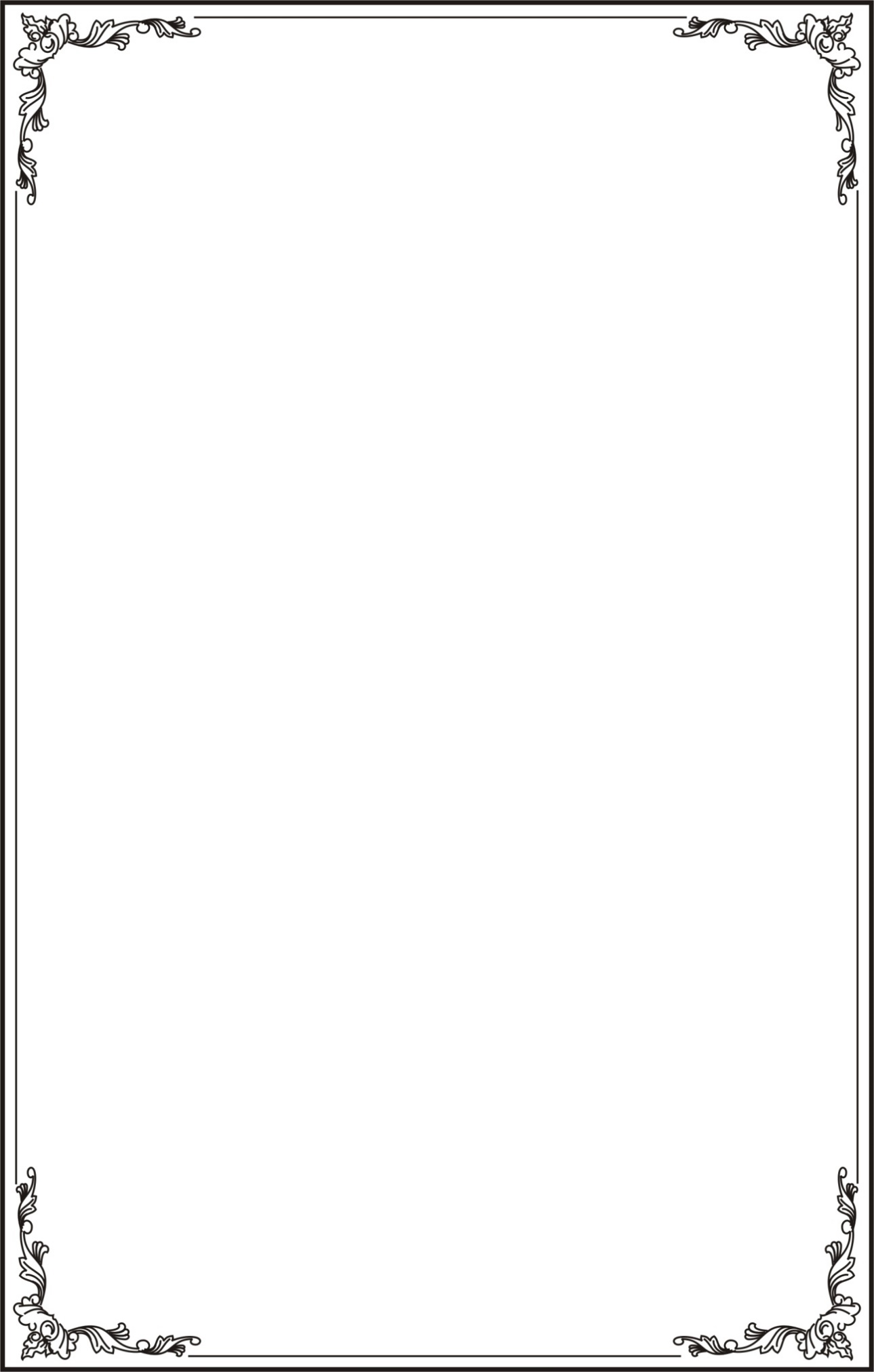
Logo, icon

Description automatically generated

**môn học: Cấu trúc dữ liệu và giải thuật**

**TIỂU LUẬN cuối kỳ**

**môn học: TƯ TƯỞNG HỒ CHÍ MINH**

**TÓM TẮT CHƯƠNG 10: aDDITION MACHINES**

**Sinh viên thực hiện**

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 1 | Lê Minh Kha | 21110890 |
| 2 | Trần Quý Thương | 21110672 |
| 3 | Nguyễn Diệu Hương | 21110489 |

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT**

**THÀNH PHỐ HỒ CHÍ MINH**

**KHOA: CÔNG NGHỆ THÔNG TIN**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT**

**THÀNH PHỐ HỒ CHÍ MINH**

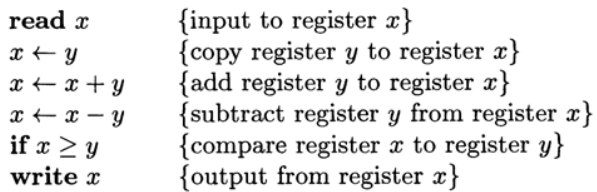
**KHOA: CHÍNH TRỊ VÀ LUẬT**

*Tp. Hồ Chí Minh, tháng 12 năm 2022*

*Tp. Hồ Chí Minh, tháng 11 năm 2022*

**1.Khái niệm Addition Machines:**

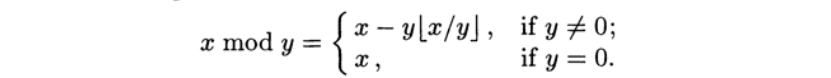
* Addition machines (máy cộng) là một thiết bị tính toán có số lượng thanh ghi hữu hạn, chỉ thực hiện được 6 phép tính cơ bản:



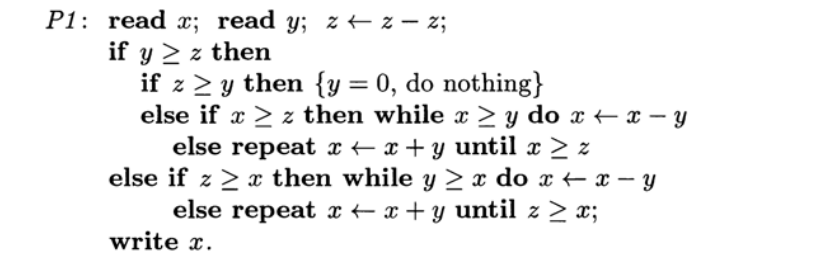
* Khi máy cộng thao tác trên các số nguyên thì ta nói máy cộng trên là máy cộng số nguyên, khi nó thao tác trên số thực thì ta nói nó là máy cộng số thực.
* Máy cộng không áp dụng một cách khuôn mẫu cho một bài toán xác định mà nó được thể hiện qua các ví dụ. Sau đây chúng ta sẽ đi xây dựng các phép toán, thuật toán chỉ dựa trên máy cộng.

**2. Remainder:**

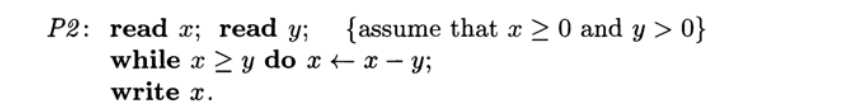
Đến với ví dụ đầu tiên, ta xem xét công thức tính số dư như sau:



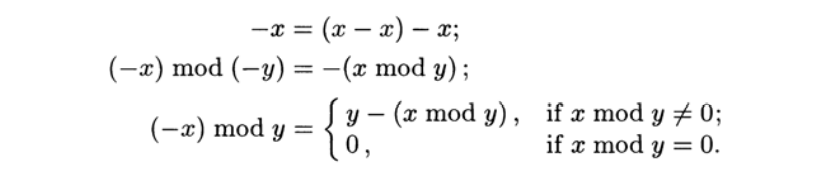
Với công thức này ta có thể dễ dàng định nghĩa trên addition machine như sau:



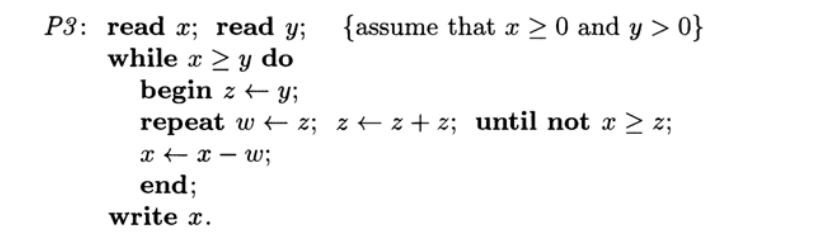
Đoạn chương trình P1 đã xử lý mọi trường hợp về dấu của x và y, trong trường hợp đặc biệt hơn ta có thể cho x ≥ 0, y > 0, để cho chương trình đơn giản hơn như sau:



Ta có thể chuyển đổi kết quả về đúng dấu bằng cách:

****

Chúng ta có thể cải thiện tốt hơn thuật toán này với thủ tục nhân đôi, một lần trừ đi bội số lớn hơn của y, thay vì chỉ trừ đi một lần y:



Chương trình lặp đi lặp lại phép trừ x cho 2ky. Với k bằng , do đó nó ngầm tính biểu diễn nhị phân của , từ trái sang phải. Tổng thời gian chạy của chương trình P3 có giới hạn là O(log(x/y))2 nhỏ hơn so với độ phức tạp của P1 và P2.

**3. A Fibonacci method:**

Bài toán tính số dư có thể được tính với độ phức tạp O(log(x/y)) hiệu quả như mong muốn, dùng phương pháp Fibonacci. Lý thuyết được trình bày như sau: Mỗi một số nguyên dương N được trình bày bằng tổng các số Fibonacci:

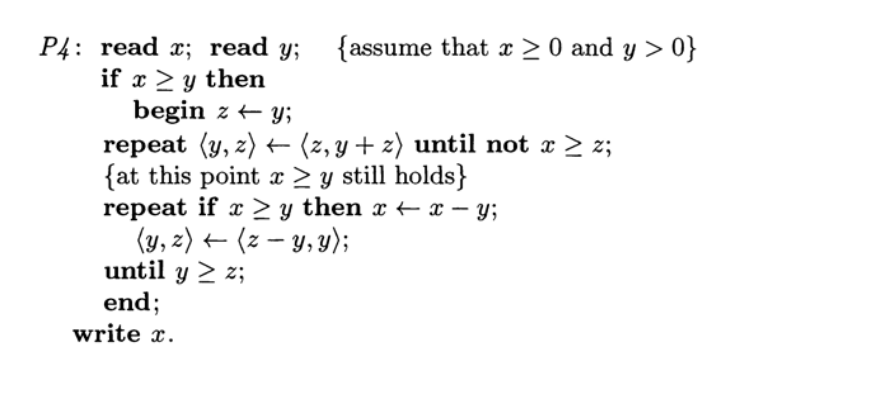
N = Fl1 + Fl2 + … + Flt

Trong đó: l1 ≥ 1 và lv+1 - lv ≥ 2 với l ≤ v < t

{Fn}: Chuỗi số Fibonacci: 0, 1, 2, 3, 5, 8, 13,…

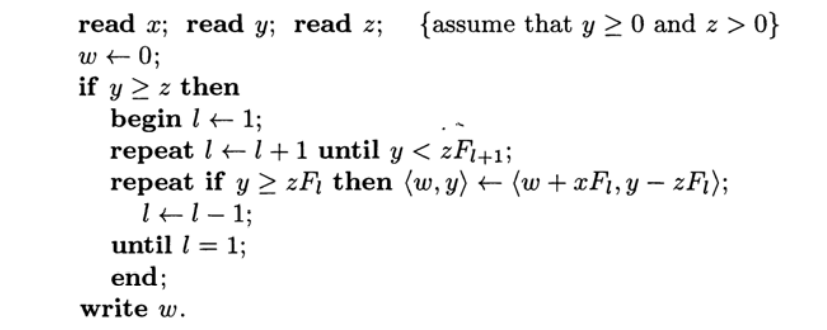
(Nguồn: E. Zeckendorf, “Représentation des nombres naturels par une somme de nombres de Fibonacci ou de nombres de Lucas”, Bulletin de la Société Royale des Sciences de Liège 41 (1972), 179 – 182)

Dựa vào đó hình thành nên thuật toán tìm số dư x mod y hiệu quả với addition machine:

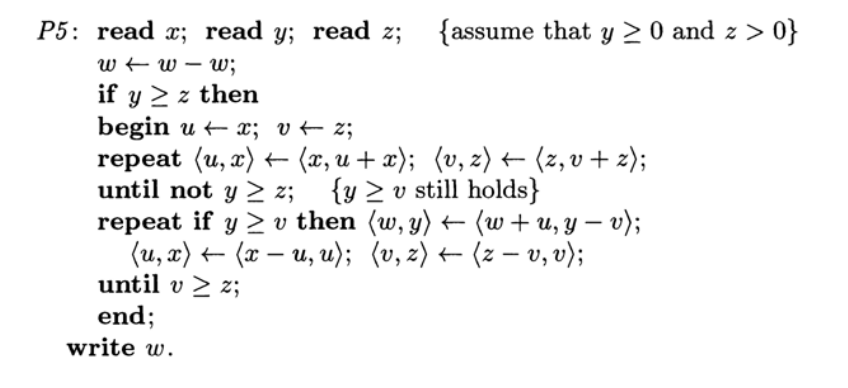


**4. Multiplication and division:**

Phần này ta sẽ tính phép tính tam phân , sử dụng cùng ý tưởng với phần trên để tính phép tính này hiệu quả. Độ phức tạp của thuật toán là O(log(y0/z0)). Lần này ta sẽ tích lũy bội số của x cũng như tìm ra Fibonacci representation của:

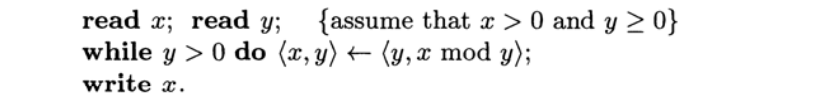


Chương trình trên addition machine thực tế yêu cầu 6 thanh ghi, bởi vì ta cần bội số Fibonacci của x và z:

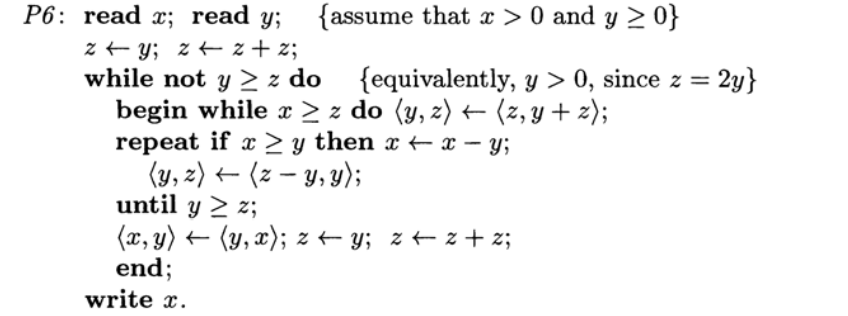


**5. Greatest common divisors:**

Phần tính ước chung lớn nhất về cơ bản ta vẫn thực hiện theo như giải thuật Euclid



Ta có thể sử dụng phương pháp tính x mod y được nêu như trên để tính gcd(x, y) trên addition machine:



Thuật toán này sẽ tính được ước chung lớn nhất của x và y chính xác khi tỉ lệ y/x là hợp lý, nếu không thuật toán sẽ lặp vô tận.

**6. Stack and Sorting:**

* Dựa trên thuật toán Euclid, ta có thể lưu trữ một ngăn xếp số nguyên dương chỉ bằng 2 thanh ghi. Với mỗi trạng thái giá trị của hai thanh ghi chúng ta sẽ xác định được một số nguyên dương. Để kết thúc quá trình nhận vào số nguyên dương cho ngăn xếp, nó sẽ được theo sau bởi số 0 và số 0 có nhiệm vụ giúp thể hiện sự kết thúc ngăn xếp và không thuộc phần tử của ngăn xếp.
* Thuật toán mô phỏng:

Text, letter

Description automatically generated

* Độ phức tạp của thuật toán: O(m + logq1q2q3…qm).
* Thuật toán trên chỉ thao tác được trên những số nguyên dương, nhóm em sẽ mở rộng ra tập số nguyên và xây dựng cấu trúc dữ liệu stack dựa vào thuật toán trên.

**7. Exponentiation mod z: ()**

* Kết hợp giữa tính chất chia lấy dư và Fibonacci reprentation để tối ưu phép tính trên mà không cần tính .
* Code mô phỏng:

Text, letter

Description automatically generated

* Độ phức tạp của thuật toán: O(logy.logz + log(x/z))
* Áp dụng tính chất toán học: để tăng tốc độ tính toán.
* Ta có thể bỏ qua Fibonacci representation để có thể tính dễ dàng hơn nhưng chấp nhận thời gian thực thi lâu hơn (sẽ trình bày trong phần ứng dụng).

**8. Further Restriction:**

* Tối giản máy cộng hơn nữa, lượt bỏ đi một số phép toán cơ bản bởi có thể thực hiện nó bằng nhiều phép tính cơ bản khác kết hợp.
* A picture containing text

  Description automatically generatedThay thế phép gán:
* Thay thế phép so sánh x y bởi:

Text, letter

Description automatically generated

* Thay thế phép cộng, chỉ bằng những phép trừ kết hợp:

A picture containing text, opener

Description automatically generated