**CHAPTER 10: ADDITION MACHINE**

**Lớp 09**

**Nguyễn Tấn Phát - 21110911**

**Trần Thành Hòa - 21110464**

**Nguyền Hồng Khoa – 21110510**

**1. POSTER**

Calendar

Description automatically generated with medium confidence

**II. SƠ ĐỒ THUẬT TOÁN VỀ BÀI BÁO**

**1. Tổng quan về Addition Machine**

Nó có thể tính gcd(x, y) một cách hiệu quả với chỉ O(logxy) những phép cộng và phép trừ khi 3 thanh ghi số học có sẵn nhưng có khi chỉ có 2. Một số chức năng khác giống như xy chia lấy dư cho z, thì vẫn có thể tính toán một cách hiệu quả trên số lượng ít thanh ghi nhỏ, sử dụng chỉ phép cộng, phép trừ và phép so sánh.  
Addition Machine là một thiết bị tính toán với giới hạn số lượng thanh ghi, được giới hạn theo 6 hoạt động sau:

|  |  |
| --- | --- |
| read x  x ← y  x ← x + y  x ← x - y if x ≥ y  write x | {truyền giá trị cho x} {sao chép giá trị y sang x} {cộng giá trị của x và y}  {trừ giá trị của x cho y} {so sánh x với y} {xuất ra giá trị của x} |

Nội dung thanh ghi được giả sử thuộc về một tập A đã cho, cái mà là một nhóm thêm vào của những số thức. Nếu A là tập hợp của tất cả các số nguyên, chúng tôi gọi thiết bị này là một máy cộng số nguyên; nếu A là tập hợp của tất cả của tất cả các số thực, chúng tôi gọi thiết bị này là một máy 1 máy cộng số thực.  
 Chúng tôi sẽ xem xét cách một máy cộng số nguyên hiệu quả có thế làm các chức năng như nhân , chia, ước chung lớn nhất, lũy thừa và sắp xếp.

Chúng tôi sẽ cẫn cho thấy bất kì máy cộng với ít nhất 6 thanh ghi có thể tính toán tử bậc c x[y/z] với hiệu quả hợp lý, nhận được x, y, z A với z 0.

**2. Các ứng dụng của Addition Machine**

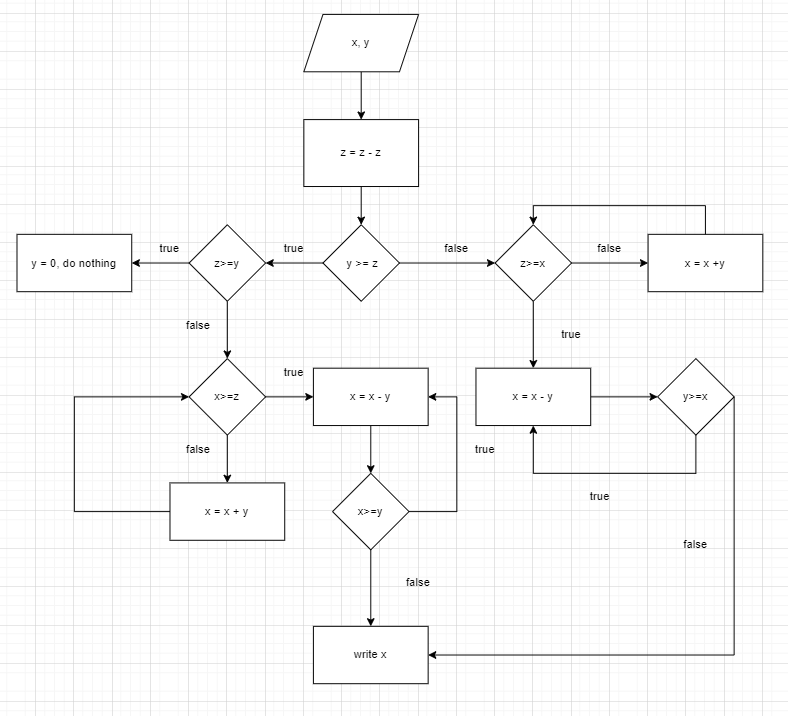
**2.1. Remainder**

Yêu cầu: Tính giá trị của x mod y.

***Input:*** Hai số nguyên x, y

***Output:*** Số dư khi chia x cho y

***Lưu đồ thuật toán:***



***Mô tả thuật toán:***

B1: Nhập x,y; gán z = z - z.

B2: So sánh y>=z?:

(1)Nếu y>=z đúng,thì so sánh z>=y? nếu z>=y đúng: gán y=0, dừng

nếu z>=y sai :So sánh x>=z? nếu x>=z đúng, khi nào x>=y còn đúng thì x = x - y

nếu x>=z sai, x = x + y cho đến khi x >= z đúng

(2)Nếu y >= z sai, thì so sánh z >= x? nếu z >= x đúng, khi nào y >= x còn đúng thì x = x - y nếu z >= x sai, x = x + y cho đến khi z >= x

B3: Ghi lại giá trị x sau khi kết thúc chương trình.

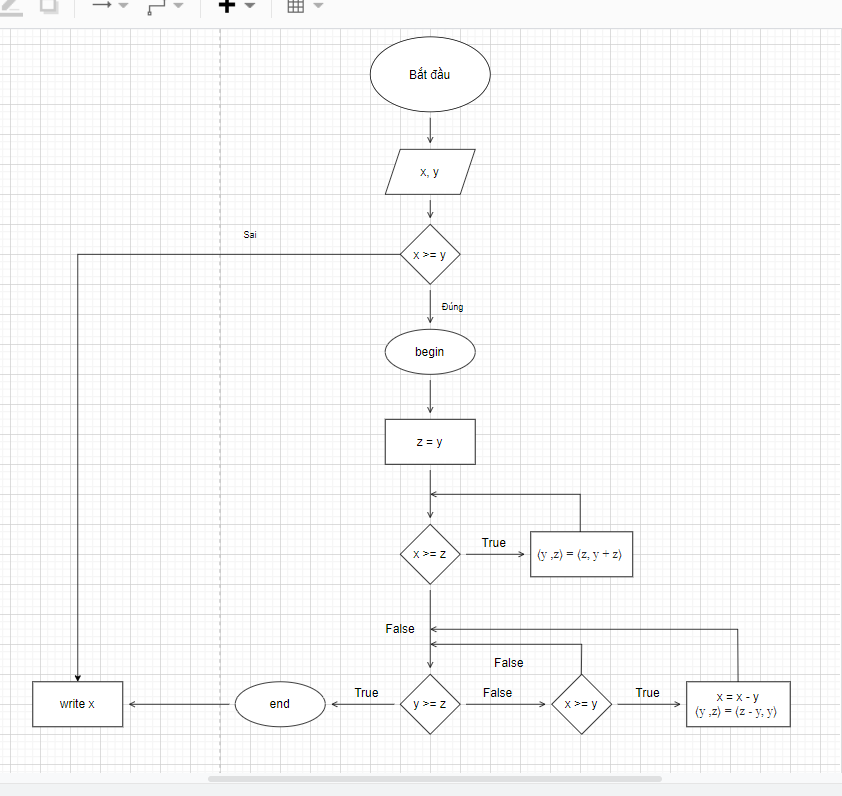
**2.2. Fibonacci Method**

Yêu cầu: Tính giá trị của x mod y nhưng với độ phức tạp nhỏ hơn Remainder

***Input:*** Hai số nguyên x, y

***Output:*** Số dư khi chia x cho y

***Lưu đồ thuật toán:***



***Mô tả thuật toán:***

B1: Nhập x, y

B2: So sánh x >= y(1)

Nếu (1) Đúng

B3: Gán z = y

B4: So sánh x >= z(2)

Nếu (2) Đúng

B5: Gán ⟨y ,z⟩ = ⟨z, y + z⟩. Quay lại bước 4

Nếu (2) Sai

B6: So sánh y >= x(3)

Nếu (3) Sai

B7: So sánh x >= y(4)

Nếu (4) Sai

B8: Quay lại B6.

Nếu (4) Đúng

B9: Gán x = x - y, ⟨y ,z⟩ = ⟨z - y, y⟩. Quay lại B6

Nếu (3) Đúng

B10: Xuất x

Nếu (1) Sai

B11: Tới B10

**2.3. Multiplication and Division**

Yêu cầu: Thực hiện phép tính x nhân với phần nguyên y chia z

***Input:*** Ba số nguyên x, y, z

***Output:*** Kết quả của phép tính x[/y]

***Lưu đồ thuật toán:***

Chart

Description automatically generated***Mô tả thuật toán:***

B1: Nhập x, y, z.

B2: Gán =0.

B3: Nếu y z thì thực hiện bước 4, ngược lại thì thực hiện bước 10.

B4: Gán l = 1.

B5: Gán l = l + 1.

B6: Nếu y thì thực hiện bước 7, ngược lại thực hiện bước 5.

B7: Nếu y thì gán = .

B8: Gán l = l – 1.

B9: Nếu l = 1 thì thực hiện bước 10, ngược lại thì trở lại thực hiện bước 7.

B10: In ra giá trị của , kết thúc chương trình.

**2.4. Greatest Common Divisors**

Yêu cầu: Tìm ước chung lớn nhất của x và y

***Input:*** Hai số nguyên của x, y

***Output:*** Ước chung lớn nhất của x và y

***Lưu đồ thuật toán:***

Chart, radar chart

Description automatically generated

***Mô tả thuật toán:***

B1: Nhập vào x, y ( với x > 0, y 0).

B2: Gán z = y.

B3: Gán z = z + z.

B4: Nếu y z thì thực hiện bước 12, ngược lại thực hiện bước 5

B5: Nếu x z thì gán x = x – y.

A picture containing text

Description automatically generatedB6: Nếu x y thì gán x = x – y.

B7: Gán = .

A picture containing text, night sky

Description automatically generatedB8: Nếu y z thì thực hiện hiện bước 9, ngược lại thì trở lại thực hiện bước 6.

B9: Gán = .

B10: Gán z = y.

B11: z = z + z.

B12: In ra giá trị của x, kết thúc chương trình.

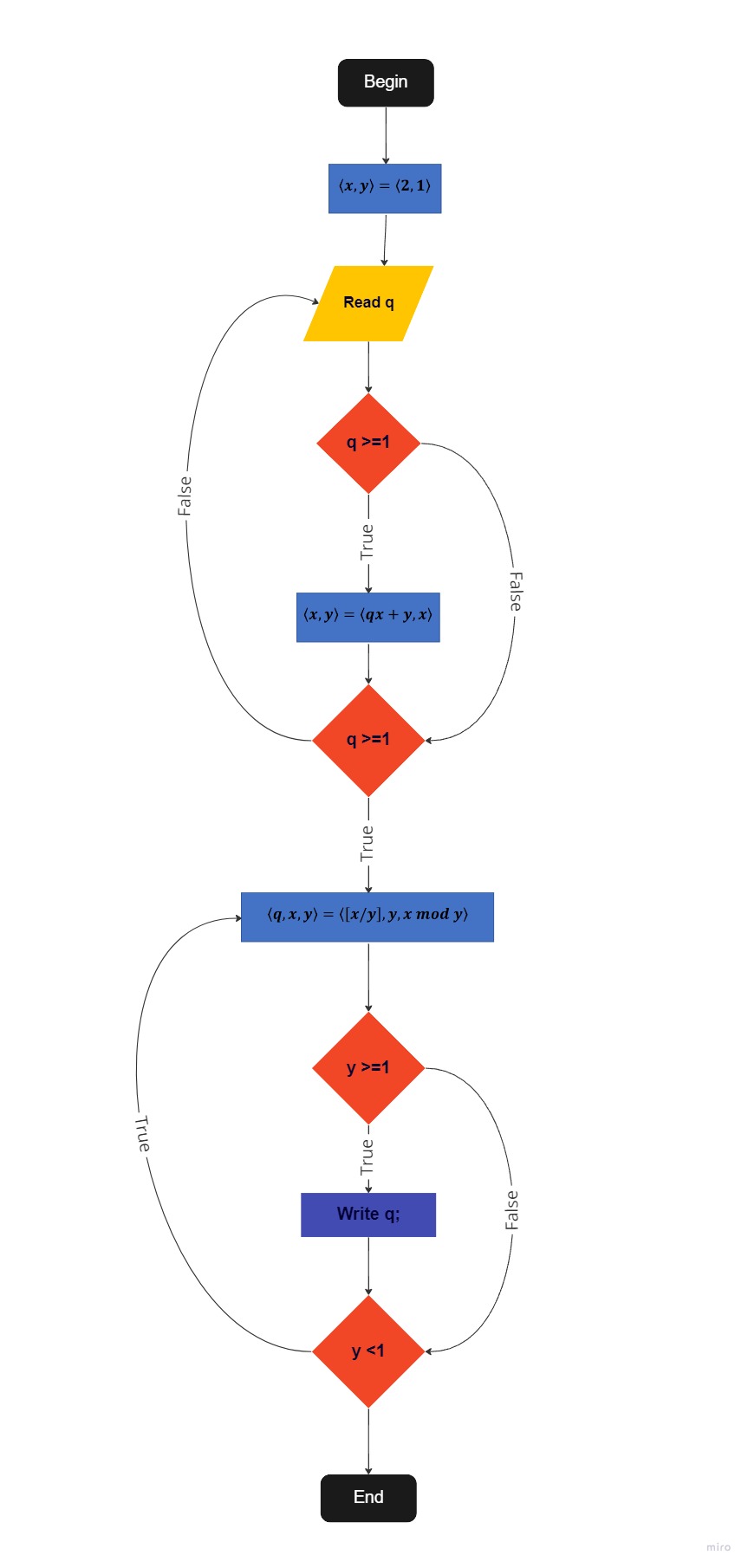
**2.5. Stack and Sorting**

Yêu cầu: Đảo ngược chuỗi số nguyên dương

***Input:*** Chuỗi số nguyên q

***Output:*** Chuỗi số nguyên q được sắp xếp theo thứ tự ngược lại

***Lưu đồ thuật toán:***



***Mô tả thuật toán:***

B1: Gán x=2, y=1;

B2: Nhập q.

B3: Nếu như q ≥1 thì gán x=q\*x+y, y=x;

B4: Nếu q <1 thì quay lại B2, ngược lại tới B5

B5: Gán q = [x/y], x = y, y = x mod y;

B6: Nếu y ≥1 thì xuất ra q.

B7: Nếu y <1 thì quay lại B5, ngược lại thì dừng lại.

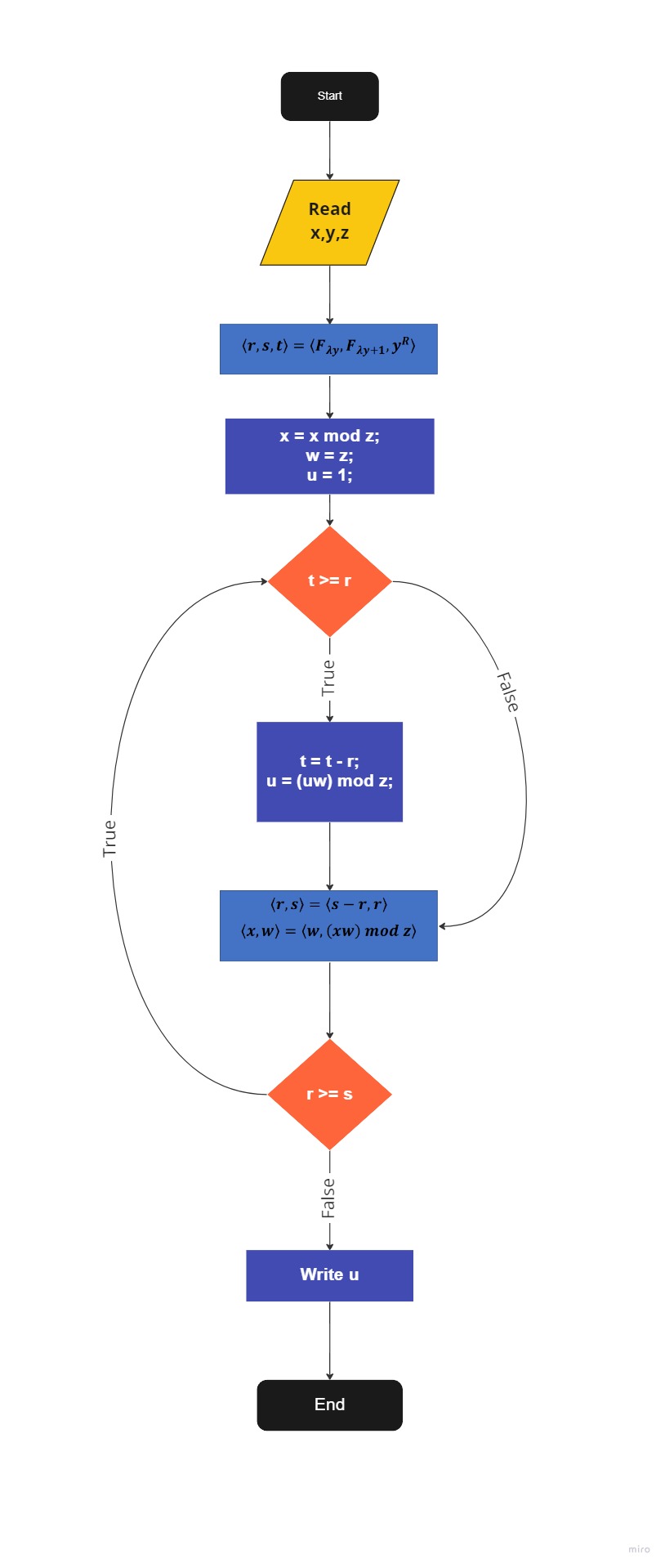
**2.6. Exponentiation mod z**

Yêu cầu: Tìm số dư của xy mod z

***Input:*** Ba số nguyên x,y,z (x ≥ 0, y > 0, z > 0).

***Output:*** Số dư của phép tính.

***Lưu đồ thuật toán:***



***Mô tả thuật toán:***

B1: Nhập x, y, z

B2: Gán

B3: Gán x = x mod z, w = z, u =1

B4: Nếu t ≥ r thì gán r = s – r, s = r và x = w, w = (xw) mod z

B5: Nếu r ≥ s thì quay lại B4, ngược lại thì xuất u.

**2.7. Further Restrictions**

Addition Machines vẫn còn khá nhiều hạn chế, điển hình là pháp gán x=y

Thay vì gán bình thường thì addition machines sẽ thực hiện

B1: Gán x = x – x

B2: Gán x = x + y;

Chúng ta cũng có thể đơn giản hóa câu lệnh ‘if x ≥ y then ’ thành

B1: Gán x = x – y;

B2: Nếu x ≥ 0 thì gán x = x + y rồi thực hiện, ngược lại x = x + y rồi thực hiện

**2.8. Lower Bound**

Một số thuật toán ứng dụng của Addition Machines ở trên có thể được chứng minh là mức độ tối ưu chỉ tăng đến một giới hạn nhất định.

Ví dụ: Chúng ta cần n(log min(x, y)) phép cộng để tính tích xy; chúng ta không thể tính bất kỳ số nào lớn hơn 2kmax(x, y) với k phép cộng.