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

CS112.O23

Thuật toán lũy thừa nhanh

Thiết kế theo phương pháp chia để trị để tính a^x

(Divide And Conquer - DAC)

Giáo viên hướng dẫn: Nguyễn Thanh Sơn
Họ và tên: Nguyễn Trần Khương An
MSSV: 22520026

1 Giới thiệu bài toán

Thuật toán lũy thừa nhanh là phương pháp hiệu quả để tính toán lũy thừa của một số nguyên a với một số nguyên dương x lớn. Báo cáo này thực hiện phương pháp chia để trị bằng đệ quy và không đệ quy để giải quyết vấn đề tính lũy thừa nhanh.

1.1 Xác định yêu cầu

- Input: Đầu vào của bài toán là hai số nguyên a và x, trong đó a là số cần tính lũy thừa, và x là số mũ.
- Output: a^x.
- Thuật toán: Sử dụng phương pháp chia để trị để giải quyết bài toán tính lũy thừa nhanh.
- Yêu cầu: Giải quyết bằng đệ quy và không đệ quy

1.2 Thuật toán chia để trị (DAC)

Phương pháp chia để trị chia bài toán thành các bài toán con nhỏ hơn, giảm bài toán lớn thành các bài toán nhỏ hơn và dễ giải quyết hơn. Sau đó, kết hợp các kết quả từ các bài toán con để đạt được kết quả cuối cùng.

- Ưu điểm:
  - Dễ dàng hiểu và triển khai.
- Nhược điểm:
  - Đòi hỏi nhiều bộ nhớ hơn so với phương pháp tham lam.
  - Có thể gặp phải vấn đề về tràn số khi tính toán lũy thừa với số nguyên rất lớn.

2 Ý tưởng

2.1 Đệ quy

Chia số mũ x thành 2 phần:

- Nếu số mũ x bằng 0, trả về 1.
- Nếu x là số chẵn, tính lũy thừa bằng cách tính bình phương của nửa lũy thừa và nhân lại với chính nó.
- Nếu x là số lẻ, tính lũy thừa bằng cách tính nửa lũy thừa của số mũ lẻ và nhân với chính nó, sau đó nhân với cơ số a.

Algorithm 1 Tính lũy thừa nhanh bằng đệ quy

function POWER(a, x)
  if $x = 0$ **then**
    return 1
  end if
  if x chia hết cho 2 **then**
    $halfpow \leftarrow$ power$(a, x//2)$
    return $halfpow \times halfpow$
  else
    $halfpow \leftarrow$ power$(a, (x-1)//2)$
    return $a \times halfpow \times halfpow$
  end if
end function

2.2 Không đệ quy

- Khởi tạo kết quả ban đầu là 1
- Vòng lặp cho đến khi x bằng 0
- Kiểm tra xem bit cuối cùng của x có phải là 1 hay không
  - Nếu là 1, nhân kết quả với a
- Tính bình phương của a
- Chia x cho 2 để giảm số lần lặp
- Trả về kết quả cuối cùng của lũy thừa

Algorithm 2 Fast Exponentiation without Recursion using Divide and Conquer

function POWER(a, x)
  $result \leftarrow 1$
  while $x > 0$ **do**
    if $x \mod 2 = 1$ **then**
      $result \leftarrow result \times a$
    end if
    $a \leftarrow a \times a$
    $x \leftarrow x \div 2$
  end while
  return $result$
end function

3 Cài đặt

Ngôn ngữ lập trình: Python

3.1 Đệ quy

```
def power(a, x):
    if x == 0:
```

```
        return 1
    if x % 2 == 0:
        halfpow = power(a, x/2)
        return halfpow * halfpow
    else:
        halfpow = power(a, (x-1)/2)
        return a * halfpow * halfpow
```

3.2 Không đệ quy

```
def power(a, x):
    result = 1
    while x > 0:
        if x % 2 == 1:
            result *= a
        a *= a
        x //= 2
    return result
```

3.3 Độ phức tạp thuật toán

Độ phức tạp của thuật toán tính lũy thừa nhanh bằng phương pháp chia để trị (DAC) ở cả hai phiên bản đệ quy và không đệ quy là $O(\log n)$, với n là số mũ.

Link github: https://github.com/kuongan/CS112/tree/main/DAC