Evaluation of Polynomials by Computer

Trương Nguyễn Thùy Trang - 21110691
Đặng Công Tuấn - 21110709
Lại Trọng Minh Trường - 21110934
Đại học Sư Phạm Kỹ Thuật TP.HCM
Đề án cuối kỳ cấu trúc dữ liệu và giải thuật
Năm học 2022 - 2023

Kết quả chính

Bài viết mô tả các phương pháp tối ưu để chuyển đổi bằng một cách nào đó thành số lượng phép nhân tối thiểu trong việc đánh giá đa thức bằng máy tính. Bài viết chỉ xem xét trường hợp đặc biệt nhất P = y^n với ba thuật toán được đưa ra lần lượt là binary method, factor method, tree method để tối ưu phép tính.

Binary method

Một phương pháp khá nổi tiếng, (xem ví dụ, Floyd [1, trang 50—51]), và có thể gọi là Binary Method, có thể được mô tả như sau:

a) Viết n dưới dạng số trong hệ thống nhị phân, ví dụ: nếu n là 13 trong số thập phân, thì n là 1101 trong nhị phân.

b) Thay thế mỗi "1" bằng SX và mỗi "0" bằng S; ví dụ: 1101 trở thành SXSXSSX.

c) Hủy SX ở đầu bên trái. Chuỗi kết quả có thể được hiểu: S = "bình phương" và X = "nhân với y". Trong ví dụ của chúng tôi, chúng tôi tính toán y^{13} với chuỗi SXSSX:

1. Lấy y;
2. Bình phương (cho y^2);
3. Nhân với y (cho y^3);
4. Bình phương (cho y^6);
5. Bình phương (cho y^{12});
6. Nhân với y (cho y^{13}).

Chúng tôi đã sử dụng 5 phép nhân, trong trường hợp này có thể được hiển thị là tối thiểu.

Factor method

Một thuật toán khác, được cho là mới, có thể được gọi là Factor Method. Ở đây chúng ta tạo ra các chuỗi S(n, m) như sau:

a) Nếu n không phải là số nguyên tố, S(n, m) = S(n/p,m)S(p, nm/p), trong đó p là thừa số nguyên tố nhỏ nhất của n.

b) Nếu n là số nguyên tố, S(n, m) = S(n - 1, m)X_m.

c) S(l, m) = chuỗi null.

Giải thích: X_m có nghĩa là "nhân với y^m" , đó là một kết quả đã được tính toán trước đó. S(n, m) là chuỗi cho phép tính của y^{mn} giả sử rằng y^m đã được tính toán. Mục tiêu là tìm S (n, 1).

Ví dụ:

$$
\begin{aligned}
S(13,1) &= S(12,1)X_1 && (1)\\
&= S(6,1)S(2,6)X_1 && (2)\\
&= S(3,1)S(2,3)X_6X_1 && (3)\\
&= S(2,1)X_1X_3X_6X_1 && (4)\\
&= X_1X_1X_3X_6X_1 && (5)
\end{aligned}
$$

Một cách khác để mô tả thuật toán này là:

a) Phân tích n thành các số nguyên tố, n = $p_1p_2...p_r$.

b) Nếu n là số nguyên tố, hãy tính $(y^{n-1})y$

c) Nếu n là hợp số, hãy tính $(...((y^{p1})^{p2})...)^{pr}$

Tree method

Tree Method - phương pháp phát triển một cái cây, là một trong những kỹ thuật tốt nhất trong các kỹ thuật xử lý danh sách. Bắt đầu ở cấp độ 0 với một nút duy nhất có nhãn 1. Để đến cấp độ tiếp theo khi một cấp độ được hoàn thành, hãy xử lý các nút ở cấp độ trước đó từ trái sang phải. Đối với mỗi nút, hãy thử thêm từng giá trị phía trên nút này, liên tiếp từ trên cùng. Nếu thu được bất kỳ giá trị mới nào, chúng sẽ trở thành các nút phân nhánh sang cấp độ tiếp theo. Các giá trị trùng lặp thu được sẽ bị loại bỏ.

Ví dụ: chúng ta có thể phát triển ba cấp độ như trong Hình 1. Loại bỏ các giá trị là trùng lặp, chúng ta nhận được Hình 2. Thực hiện các bổ sung thử nghiệm từ trên xuống dưới dẫn đến một cây tốt hơn nhiều so với việc tạo ra chúng từ dưới lên trên. Tuy nhiên, phương pháp từ dưới lên trên thuận tiện hơn nhiều để lập trình và nó có thể được sử dụng nếu các nhánh mới từ nút được gắn từ phải sang trái. Cây bắt đầu trông như trong Hình 3.

FIGURE 1.

FIGURE 2.

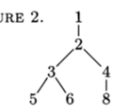
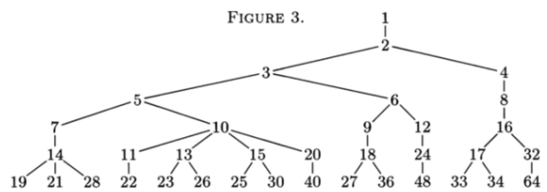


FIGURE 3.



So Sánh giữa các phương pháp

Để so sánh hai phương pháp đầu, Binary Method sử dụng r + s - 1 phép nhân, trong đó r là số nguyên lớn nhất nhỏ hơn hoặc bằng $log_2 n$ và s là số thứ nhất trong biểu diễn nhị phân của n. Factor Method sử dụng phép nhân M_n, trong đó:

$$M_n = \begin{cases} M_{n-1} + 1, & \text{if } n \text{ is prime;} \\ M_r + M_s, & \text{if } n = rs \text{ is composite.} \end{cases}$$

Factor Method tốt hơn một chút so với Binary Method nói chung:
Đối với n ≤ 150, có:

- 93 trường hợp hai phương pháp bằng nhau,
- 32 trường hợp Factor method là tốt hơn 1 phép toán,
- 16 trường hợp Binary method là tốt hơn 1 phép toán,
- 8 trường hợp Factor method là tốt hơn 2 phép toán,
- 1 trường hợp (n = 129) Binary method là tốt hơn 2 phép toán.

Các trường hợp nhỏ nhất mà Factor Method vượt trội là n = 15, 27, 30, 31* , 39, 45. Các trường hợp nhỏ nhất mà Binary Method vượt trội là n = 33, 49, 65, 66, 67, 69.

Nhưng có những trường hợp chúng ta có thể làm tốt hơn cả hai phương pháp. Đối với n ≤ 70, có 5 trường hợp được biết đến mà điều này là đúng. (Chúng là n = 23, 43, 46, 47, 59).

Phương pháp thứ ba, là kỹ thuật tốt nhất mà tác giả biết đến, là Tree Method.

Hướng phát triển

Chúng tôi chỉ tập trung vào tổng số phép nhân là tiêu chí của sự xuất sắc trong cuộc thảo luận này.

Binary method: Nếu n là một biến chưa biết, Binary method là ưu việt hơn. Trong thực tế, phương pháp đó rất phù hợp để kết hợp trong phần cứng của một máy tính nhị phân, như một toán tử lũy thừa.

Factor method: có giá trị khi n lớn và một sự ứng dụng đòi hỏi phải tính toán thường xuyên y^n.

Tree method: Nếu y là một số thực dấu phẩy động, tất nhiên có một điểm của quy luật hiệu suất giảm dần (point of diminishing returns) khi n dần lớn, vì cuối cùng chúng ta sẽ tốt hơn bằng cách lấy logarit và lũy thừa. Vì Tree method sử dụng r phép nhân ở cấp độ r, có thể ngừng tạo cây ở một mức nhất định; sau đó chúng ta có tập hợp tất cả các giá trị "thú vị" của n. Ví dụ, cây trong Hình 3 trong bài báo cho thấy tất cả n mà người ta biết rằng y^n cần ít nhất là 6 phép nhân.

Tài liệu

[1] obert W. Floyd, *"An algorithm for coding efficient arithmetic operations,"* Communications of the ACM 4 (1961), 42—51.

[2] . M. Ostrowski, *"On two problems in abstract algebra connected with Horner's rule,"* Studies in Mathematics and Mechanics Presented to Richard von Mises (New York: Academic Press, 1954),

[3] ohn Todd, A Survey of Numerical Analysis (New York: McGraw— Hill, 1962), 3-4.