



Chương 3: A One Way, Stackless Quicksort Algorithm

Thái Bảo An - 21110365
Phan Hiếu Thuận - 21110445
Nguyễn Dương Thế Vĩ - 21110728
Đồ án cuối kì Cấu trúc dữ liệu và giải thuật. Năm học: 2022- 2023

Nội dung

Bài viết mô tả một kĩ thuật sắp xếp mới được trình bày từ [1]. Thuật toán này tương tự thuật toán sắp xếp "*Quicksort*" mà chúng ta đã biết, nhưng không dùng đệ quy làm chương trình được tối ưu hơn. Kĩ thuật mới yêu cầu các key được sắp xếp phải là các số nguyên dương.

Hoare’s Quicksort

Hoare "*Quicksort*" [2] được phát minh lần đầu bởi C.A.R.Hoare vào năm 1960, trong nhiều năm qua nó đã là một thuật toán được nghiên cứu và sử dụng rộng rãi nhất trong các thuật toán sắp xếp.

Ý tưởng của thuật toán này dựa trên phương pháp chia để trị, chia mảng thành hai phần bằng cách so sánh từng phần tử của mảng với một phần tử được gọi là phần tử chốt (*Pivot*). Một mảng bao gồm các phần tử nhỏ hơn hoặc bằng phần tử chốt và một mảng gồm các phần tử lớn hơn phần tử chốt.

Quá trình phân chia này diễn ra cho đến khi độ dài của các mảng con đều bằng 1. Với phương pháp đệ quy ta có thể sắp xếp nhanh các mảng con sau khi kết thúc chương trình ta được một mảng đã sắp xếp hoàn chỉnh.

Thuật toán "*Quicksort*" có nhiều ưu điểm như là có độ phức tạp nhỏ hơn các thuật toán sắp xếp đơn giản, tốc độ xử lý tương đối nhanh. Trong một số trường hợp, *Quicksort* là thuật toán có tốc độ tốt nhất Thông dụng, được áp dụng nhiều trong lập trình, trong thư viện của các ngôn ngữ lập trình như C++, Java, C . . . Có thể ứng dụng vào xử lý dữ liệu lớn.

Tuy vậy "*Quicksort*" vẫn còn một số hạn chế, thuật toán không có tính ổn định, không có tính thích ứng, dễ ảnh hưởng bởi dữ liệu đầu vào tồn không gian bộ nhớ hơn so với các thuật toán sắp xếp đơn giản tư tưởng và giải thuật khá phức tạp khó khăn trong việc lựa chọn phần tử làm chốt trong phân hoạch. Việc lựa chọn có thể ảnh hưởng rất nhiều đến hiệu suất của thuật toán tuy nhiên ta không thể đưa ra lựa chọn tối ưu nhất.

Vì "*Quicksort*" còn nhiều nhược điểm nên có nhiều biến thể của nó được ra đời [3]. Trong đó có một thuật toán sắp xếp mới đơn giản đến kinh ngạc được gọi là *OSORT*.

Stackless Quicksort - OSORT

Thuật toán sắp xếp OSORT yêu cầu các Record được đưa vào phải có các key là các số dương và phần tử cuối cùng trong mảng các Record phải là một số âm. OSORT chỉ duyệt qua mảng một lần và phần tử chốt (*Pivot*) được chọn từ trái sang phải trong mảng , thay vì được chọn một cách ngẫu nhiên (burning the candle at both ends) như các thuật toán Quicksort cổ điển.

Code của OSORT bằng ngôn ngữ lập trình Pascal:

```
procedure OSORT(N: integer);
label 3;
var L: 1..M; {progress indicator}
    J: 1..M; {end of current pivoting region}
    I: 1..M; {gap in current pivoting region}
    P: entry; {the pivot record}
begin R[N+1].key := -1;
for L := 1 to N do
  begin while R[L].key > 0 do
    begin J := L; I := L; P := R[L];
3:  repeat J := J + 1 until P.key > R[J].key;
    if R[J].key > 0 then
      begin R[I] := R[J]; I := I + 1;
        R[J] := R[I]; goto 3;
      end;
    P.key := -P.key; R[I] := P;
    end;
  R[L].key := -R[L].key;
end;
end;
```

Phân tích độ phức tạp của OSORT:

- Sử dụng định luật Kirchhoff và vì có đúng một key là số âm ta có thể dễ dàng tính được thời gian chạy chương trình sẽ là: $4C + 11X + 19N + 2$.

Trong đó:

- + C là số lần so sánh.
- + X là số bước sắp xếp lại.
- Giá trị trung bình C_N và X_N của C và X thỏa mãn:

$$C_N = \frac{1}{N} \sum_{1 \leq i \leq N} (N + C_{i-1} + C_{N-i}),$$
$$X_N = \frac{1}{N} \sum_{1 \leq i \leq N} (i - 1 + X_{i-1} + X_{N-i})$$

- Khi $N > 0$; $C_0 = X_0 = 0$. Nên:

$$C_N = 2(N + 1)H_N - 3N; \quad X_N = (N + 1)H_N - 2N.$$

- Do đó thời gian trung bình chạy chương trình là

$$19(N + 1)H_N - 15N + 2 \approx 19N \ln N - 4.033N + O(\log N).$$

Qua đó ta có thể thấy được thời gian chạy của OSORT tương đối nhanh hơn Quicksort, trong khi đó độ phức tạo trung bình của Quicksort là $O(N \log N)$, một số trường hợp xấu nhất độ phức tạp của nó có thể lên tới $O(N^2)$.

So sánh Quicksort và OSORT

Quicksort:

- Phần tử chốt (Pivot) ban đầu được chọn một cách tùy ý, nên thời gian chạy của thuật toán phụ thuộc vào phần tử chốt được chọn ban đầu.
- Phân hoạch duyệt qua mảng nhiều lần, dùng đệ quy.
- Độ phức tạp thay đổi phụ thuộc vào phần tử chốt được chọn. Độ phức tạo trung bình là $O(N \log N)$, một số trường hợp xấu nhất độ phức tạp của nó có thể lên tới $O(N^2)$.

OSORT:

- Phần tử chốt (Pivot) được chọn là con trỏ duyệt từ trái sang phải trong mảng
- Chỉ duyệt qua mảng một lần từ trái sang phải
- Độ phức tạp tương đối không thay đổi nhiều.

Tài liệu

[1] Huang Bing-Chao and Donald E. Kuthm *A one-way, stackless quicksort algorithm, BIT*, 26 (1986), 127-130.
[2] C. A. R. Hoare, "*Quicksort*", *The Computer Journal* , 5 (1962), 10-15.
[3] Donald E. Knuth, *The Art of Computer Programming*, Volume 3: *Sorting and Searching* Reading, Mass.: Addison-Wesley, 1975

