ĐÁNH GIÁ MỘT SỐ THUẬT TOÁN THÔNG DỤNG

Phạm Thế Bảo
Khoa Toán – Tin học
Trường Đại học Khoa học Tự nhiên Tp.HCM

Tìm kiếm tuần tự

- Xét một mảng các phần tử a[1], a[2], …, a[n]. Phần tử a[i] có khóa tìm kiếm là a[i].key, bài toán: cho trước khóa k, có tồn tại j để a[j].key bằng k hay không?

```
i=1;
found=false;
while((i≤n)&&(not found)) do
    if(a[i].key bằng k) then
        found=true;
    else
        i=i+1;
    endif
endw
```

Nếu bỏ else:

1. **Thuật toán còn đúng không?**
2. **Có tăng phép đếm (gán)?**

- Ta cần phân biệt:
  - ❖ Phép toán số học: so sánh, gán
  - ❖ Phép toán trên khóa: sao chép, so sánh
- Nếu ta thêm một phần tử a[n+1].key=k thì số phép toán sẽ tăng hay giảm?
- Viết lại thuật toán:

```
i=1;
a[n+1].key=k;
while (a[i].key khác k) do
    i = i+1;
endw
```


- Thuật toán dừng khi nào?
  - $i = n+1$ → không tìm thấy
  - $i=i_0$, với $1 \le i_0 \le n$ → tìm thấy
- Để đánh giá, ta cần tính α:
  - Tìm không thấy: $k \notin \{a[i].key / i=1..n\}$ → $\alpha=n$, gọi q là xác suất tìm không thấy.
  - Tìm thấy sẽ có xác suất là (1-q)
  - Đặt p_i là xác suất để a[i].key bằng k
  - Giả thiết a[k].key khác a[l].key nếu $k \neq l$
  - Nếu a[i].key bằng k thì $\alpha=i-1$???
  - Vậy $q+(1-q)\sum_{i=1}^{n} p_i = 1$ và có $\alpha_{\text{trung bình}} = \overline{\alpha} = \sum_{i=1}^{n} p_i(i-1)$

- Khi tìm thấy số lượng so sánh khóa:
  - Tối thiểu = 1
  - Tối đa = n
  - Trung bình = $\overline{\alpha}+1=\sum_{i=1}^{n} p_i$
- Số lần so sánh khóa trung bình cho cả hai trường hợp tìm thấy và không tìm thấy là:

$$(n+1)q+(1-q)\sum_{i=1}^{n} ip_i$$

Phạm Thế Bảo

Xem xét phân bố khóa

1. Giả sử a[i].key=i

   k được chọn ngẫu nhiên từ tập hợp 1, 2, 2, 3, 3, 3, …, $\underbrace{i, i, \ldots, i}_{\text{i lần}}$, …, $\underbrace{n, \ldots, n}_{\text{n lần}}$, n+1, n+2, …, 2n

   Tổng số khả năng có thể là:(1+2+…+n)+n=$\frac{n(n+3)}{2}$

→ Xác suất để k$\notin$ {key} là $q=\frac{n}{\frac{n(n+3)}{2}}=\frac{2}{n+3}$

Suy ra $p_i=\frac{i}{\frac{n(n+1)}{2}}=\frac{2i}{n(n+1)}$

Phạm Thế Bảo

→ Số lần so sánh khóa trung bình là:

$$= (n+1)\frac{2}{n+3} + \left(1 - \frac{2}{n+3}\right)\left(\sum_{i=1}^{n} i\frac{2i}{n(n+1)}\right)$$

$$= \frac{2(n+1)}{n+3} + \frac{n+1}{n+3}.\frac{2}{n(n+1)}.\frac{n(n+1)(2n+1)}{6}$$

$$= \frac{2n^2 + 9n + 7}{3(n+3)}$$

Phạm Thế Bảo

2. Giả sử dữ liệu phân bố đều → $p_i = \frac{1}{n}, \forall i = \overline{1..n}$
   – Số lần so sánh khóa trung bình khi tìm thấy:

$$\sum_{i=1}^{n} ip_i = \frac{1}{n}\sum_{i=1}^{n} i = \frac{n+1}{2}$$

3. Giả sử có phân bố khóa như sau:

$p_1 = c \qquad p_2 = \frac{c}{2}$

$p_3 = \frac{c}{2^2} \qquad \ldots$

$\ldots \qquad p_n = \frac{c}{2^{n-1}}$

ta có $\sum_{i=1}^{n} p_i = 1 = c\sum_{k=1}^{n} \frac{1}{2^{k-1}} = c\frac{1-\left(\frac{1}{2}\right)^n}{1-\frac{1}{2}} = 2c\left[1-\left(\frac{1}{2}\right)^n\right]$

$$\Rightarrow c = \frac{1}{2\left[1-\left(\frac{1}{2}\right)^n\right]}$$

Phạm Thế Bảo

- Số lần so sánh khóa trung bình khi tìm thấy:

$$\sum_{i=1}^{n} ip_i = \sum_{i=1}^{n} i\frac{c}{2^{i-1}} = c\sum_{i=1}^{n}\frac{i}{2^{i-1}}$$

$$\text{xét } f(x)=\sum_{i=1}^{n} ix^{i-1} = \left(\sum_{i=1}^{n} x^{i}\right)' = \left[\frac{x(1-x^n)}{1-x}\right]'$$

$$= \frac{nx^{n+1}-(n+1)x^n+1}{(1-x)^2}$$

$$\Rightarrow \sum_{i=1}^{n} ip_i = cf\left(\frac{1}{2}\right) \text{ với c được tính như trên}$$

$$\Rightarrow \sum_{i=1}^{n} ip_i = \frac{2-\dfrac{n+2}{2^n}}{1-\dfrac{1}{2^n}} \quad \Rightarrow \text{ khi n đủ lớn } (n\to\infty) \Rightarrow 2$$

Phạm Thế Bảo

- Nếu thuật tóan phân bố như trên thì độ phức tạp của thuật toán là hằng (nhỏ).
- Do những phần tử thường xuyên được gặp nhất được sắp ở đầu, những phần tử ở đầu có xác suất gặp cao hơn các phần tử càng về sau, tỷ lệ này giảm dần rất nhanh theo hệ số 2.

Ví dụ: ứng dụng trong tổ chức dữ liệu của hệ cơ sở dữ liệu Oracle

cuu duong than cong. com

Phạm Thế Bảo

4. Xem xét một phân bố khác như sau:

$$p_1 = \frac{c}{1} \qquad p_2 = \frac{c}{2}$$

$$p_3 = \frac{c}{3} \qquad \ldots$$

$$\ldots \qquad p_n = \frac{c}{n}$$

$$\text{ta có } \sum_{i=1}^{n} p_i = 1 = c\sum_{k=1}^{n} \frac{1}{k} = cH_n$$

$$\text{mà } H_n = 1 + \frac{1}{2} + \ldots + \frac{1}{n} = O(\ln n)$$

- Lúc đó số phép so sánh khóa trung bình trong trường hợp tìm thấy là

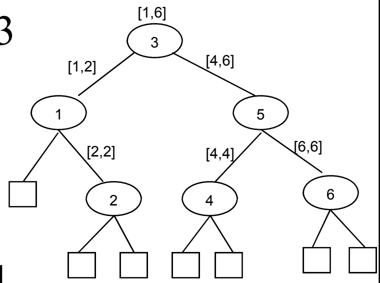
$$\sum_{i=1}^{n} ip_i = \sum_{i=1}^{n} i\frac{\frac{1}{H_n}}{i} = \frac{n}{H_n} \approx \frac{n}{\ln n}$$

Tìm kiếm nhị phân

- Cho mảng n phần tử thỏa a[1].key<…<a[n].key
- Tổng quát, ta sẽ xét từ a[l] đến a[r], với l≤r:
  - Tính m=[(l+r)/2]
  - Nếu a[m].key bằng k → dừng
  - Nếu a[m].key nhỏ hơn k, quá trình tìm kiếm lặp lại cho bên phải, nghĩa là l=m+1
  - Nếu a[m].key lớn hơn k, quá trình tìm kiếm lặp lại cho bên trái, nghĩa là r=m-1
- Thay vì tính như trên, ta tính m=(a[l].key+a[r].key)/2 → chuyện gì?

- Ví dụ: xét n=6, m=(1+6)/2=3
  - Nếu k∈{khóa} thì thuật toán dừng ở đâu?

    Số lần lặp trung bình ≈



$$\frac{1x1+2x2+3x3}{6}=\frac{14}{6}$$

- Nếu k∉ {khóa} thì thuật toán dừng ở đâu?

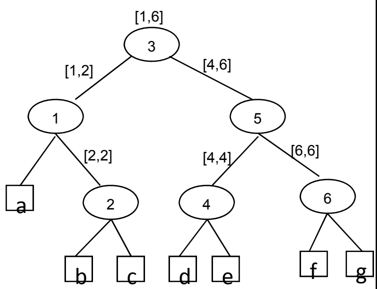
  Số lần lặp trung bình ≈

a∈(-∞,a[1].key)

b∈(a[1].key,a[2].key) c∈(a[2].key,a[3].key)

d∈(a[3].key,a[4].key) e∈(a[4].key,a[5].key)

f∈(a[5].key,a[6].key) g∈(a[5].key,+ ∞)



$$\frac{1x2+6x3}{7}=\frac{20}{7}$$

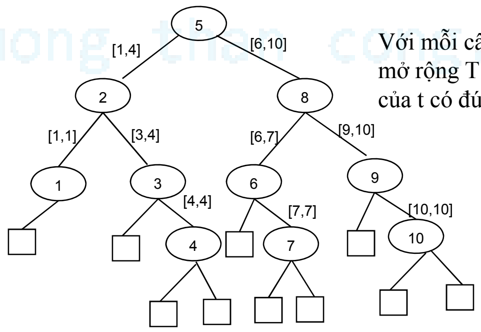
```
Thuật toán:
l=1;
r=n;
idx=-1;
while (l≤r) do
        m=[l+r]/2;
        if(a[m].key==k) then
                idx=m;
                l=r+1;
        else
                if(if(a[m].key<k) then
                        l=m+1;
                else
                        r=m-1;
                endif
        endif
endw
```

Phạm Thế Bảo

- $\beta=1$ khi $k\in$ {khóa} và $\beta=0$ khi $k\notin$ {khóa}
- Có $2\alpha-\beta$ so sánh khóa
- Ta nhận thấy: $1\leq \alpha \leq \log_2 n$
- Ước lượng chính xác giá trị trung bình của α:

Ta nhận thấy có thể biểu diễn theo cây, với định nghĩa quy nạp cho cây: với đoạn [l,r] cây có gốc là m=[(l+r)/2] và cây con trái được xây dựng với đoạn [l,m-1] và cây con phải được xây dựng với đoạn [m+1,r].

Ví dụ: n=10



Với mỗi cây T, ta xây dựng cây mở rộng T_1 sao cho mỗi node của t có đúng hai con

Phạm Thế Bảo

- Thuật ngữ:
  - Node trong (node tròn) của T=node của T=n
  - Node ngoài (vuông) của T=node bổ sung=N
  - Độ dài đường đi đến node x: l(x)=số cạnh từ gốc đến x.
  - Độ dài đường đi trong cây T= $\sum_{x \in \{node\ trong\}} l(x) = I(T)$

    Trở lại ví dụ trên, độ dài = 0x1+2x1+4x2+3x3=19
  - Độ dài đường đi trung bình đến mỗi node= $\frac{I(T)}{\text{số node trong}}$

    Trở lại ví dụ, =19/10=1.9
  - Độ dài đường đi ngòai = E(T) = $\sum_{x \in \{\text{node ngoài}\}} l(x)$
  - Độ dài đường đi ngòai trung bình = $\frac{E(T)}{\text{số node ngoài}}$

- Mệnh đề:
  a. Số node ngoài = số node trong +1, N=n+1
  b. E(T)=I(T)+2n
  c. Độ dài đường đi ngòai trung bình = $\frac{I(T)+2n}{n+1}$

Ví dụ trên, có E(T)=

I(T)=

E(T)=I(T)+2x10

- Nhận xét:
  - Khi tìm thấy: dừng ở node tròn x
    - β=1 và α=l(x)+1
    - $\overline{\alpha} = \dfrac{\sum_{x\in\{\text{node tròn}\}} [l(x)+1]}{n} = \dfrac{I(T)}{n} + 1$
    - Số phép so sánh khóa TB=$2\overline{\alpha} - \beta = 2\left[\dfrac{I(T)}{n} + 1\right] - 1 = \dfrac{I(T)}{n} + 1$
  - Khi không tìm thấy: dừng ở node vuông y
    - β=0 và α=l(y)
    - $\overline{\alpha} = \dfrac{E(T)}{N} = \dfrac{I(T)+2n}{n+1}$
    - Số phép so sánh khóa TB= $2\overline{\alpha} - \beta = 2\left[\dfrac{I(T)+4n}{n+1}\right]$

Phạm Thế Bảo

Sắp xếp chèn

- Có n phần tử a[1], …, a[n], ý tưởng:
  - n=1 hiển nhiên a[1] được sắp
  - Giả sử có k phần tử đầu a[1].key≤… ≤ a[k].key được sắp, ta phải tìm cách chèn a[k+1] vào đúng vị trí.

Ví dụ: n=7, có mảng: 10 2 7 9 6 1 5

Lần 1 chèn 2 trước 10

Lần 2 chèn 7 giữa 2 và 10

…

Phạm Thế Bảo

Thuật toán:

```
j=2;
while (j≤n) do
        i=j-1;
        k=a[j].key;
        r=a[j];
        while ((i>0)&&(k<a[i].key)) do
                a[i+1]=a[i];
                i=i-1;
        endw
        a[i+1]=r;
        j=j+1;
endw
```


- Xét P(j) có hai trường hợp:
  - Không tối ưu hóa biểu thức: (α_j+1) so sánh số học và (α_j+1) so sánh khóa.
  - Tối ưu:
    - i có thể giảm về 0: (α_j+1) so sánh số học và α_j so sánh khóa
    - i không thể giảm về 0: (α_j+1) so sánh số học và (α_j+1)so sánh khóa
  - Mục tiêu là xác định α_j:
    - Nhận xét mảnh có cấu trúc như sau:

      σ_{cur} =

Khóa tăng	a_j	

    - Gọi $\sigma=a_1a_2\ldots a_n$: hoán vị ban đầu
    - α_j = số phần tử bên trái a_j trong σ_{cur} mà lớn hơn a_j

      = số phần tử bên trái a_j trong σ mà lớn hơn a_j

- Vậy $\sum_{j=1}^{n}\alpha_j =$ số nghịch thế của σ

  có $\alpha_1 = 0 \Rightarrow \sum_{j=2}^{n}\alpha_j =$ số nghịch thế của σ

a. Số phép gán số học $= 1+(n-1)+\left[\sum_{j=2}^{n}(\text{gán số học P(j)})\right]+n+1$

$$= 2n-1+\sum_{j=2}^{n}\alpha_j = 2n-1+\text{số nghịch thế của } \sigma \begin{cases} \text{min}=0 \\ \text{max}=\dfrac{n(n-1)}{2} \\ \dfrac{n(n-1)}{4} \end{cases}$$

b. So sánh số học $= n+\sum_{j=2}^{n}(\alpha_j+1) = 2n-1+\text{số nghịch thế của } \sigma$

c. Sao chép khóa = n-1

Phạm Thế Bảo

d. Sao chép mẫu tin $= (n-1)+\left[\sum_{j=2}^{n}(\text{chép mẫu tin P(j)})\right]+n-1$

$$= 2n-2+\sum_{j=2}^{n}\alpha_j = 2n-2+\text{số nghịch thế của } \sigma$$

e. So sánh khóa:

- Không tối ưu $= \sum_{j=2}^{n}(\alpha_j+1) = n-1+\text{số nghịch thế của } \sigma$
- Có tối ưu:
  - a[j] là cực tiểu so với bên trái: i có thể giảm về 0
  - Ngược lại i không giảm về 0

$$= \begin{pmatrix} \sum_{j=2}^{n}\alpha_j \\ a[j] \text{ loại 1} \end{pmatrix} + \begin{pmatrix} \sum_{j=2}^{n}(\alpha_j+1) \\ a[j] \text{ loại 2} \end{pmatrix}, \text{ a[1] là loại 1, loại 1 và 2 bù nhau}$$

$$= \sum_{j=2}^{n}\alpha_j + \sum_{j=2}^{n}1$$

Phạm Thế Bảo

Vậy số phép so sánh khóa là (số nghịch thế của σ +(n- số phần tử cực tiểu bên trái))

$$\Rightarrow TB = \frac{n(n-1)}{4} + n - H_n$$

Sắp xếp chọn

- Ý tưởng: xét j=n, …, 2 chọn max trong {a[1].key, a[2].key, …, a[j].key} tại idx, đổi chỗ a[j] và a[idx].

  ví dụ: 10 2 7 9 6 1 5
  - j=n chọn idx=1 → hoán đổi
  - j=n-1 chọn idx=9 → hóan đổi
  - …

Thuật toán:

```
j=n;
while (j≥2) do
        idx=1;
        i=2;
        while (i≤j) do
                if(a[i].key>a[idx].key) then
                        idx=i;
                endif
                i=i+1;
        endw
        a[j] ←→ a[idx];
        j=j-1;
endw
```

Phạm Thế Bảo

- Đoạn P(j) là tìm khóa lớn nhất trong tập j phần tử. Ước lượng tổng chi phí trung bình của α_j như sau:

$$\sum_{j=1}^{n} p_j = (n+1)H_n - n$$

Phạm Thế Bảo