Chapter 02

ĐỆ QUY – QUAY LUI – NHÁNH CẬN

Email: anvanminh.haui@gmail.com

Nội dung

- Bài toán liệt kê
- Một số kiến thức về đại số tổ hợp
- Phương pháp sinh
- Đệ quy
- Quay lui
- Nhánh cận
- Một số bài tập

Bài toán liệt kê

- **Có một số bài toán trên thực tế yêu cầu chỉ rõ: Trong một tập các đối tượng cho trước có bao nhiêu đối tượng thỏa mãn những điều kiện nhất định.**
- **Bài toán này được gọi là bài toán đếm.**
- **Trong lớp các bài toán đếm, có những bài toán yêu cầu chỉ rõ những "cấu hình" tìm được thỏa mãn điều kiện là những cấu hình nào.**
- **Những bài toán này gọi là bài toán liệt kê.**
- **Thuật toán giải bài toán liệt kê cho phép lần lượt xây dựng được tất cả các cấu hình đang quan tâm.**

Bài toán liệt kê

- **Có nhiều phương pháp liệt kê, nhưng chúng cần đáp ứng được 2 yêu cầu sau:**
  - Không được lặp lại một cấu hình
  - Không được bỏ sót một cấu hình
- **Phương pháp liệt kê là phương kế cuối cùng để giải một số bài toán tổ hợp.**
- **Khó khăn của phương pháp liệt kê là sự bùng nổ tổ hợp dẫn tới sự đòi hỏi lớn về không gian nhớ và thời gian thực hiện chương trình.**
- **Chỉ nên dùng phương pháp liệt kê khi không còn phương pháp nào khác để tìm ra lời giải.**

Một số kiến thức về đại số tổ hợp

- **Cho S là một tập hữu hạn n phần tử và k là một số tự nhiên.**
- **Gọi X là tập các số nguyên dương từ 1 đến k: X = {1, 2, …, k}**
- **Chỉnh hợp lặp:**
  - Mỗi ánh xạ f : X → S. Cho tương ứng mỗi i ∈ X, một và chỉ một phần tử f(i) ∈ S. Được gọi là một chỉnh hợp lặp chập k của S.
  - Ví dụ: S = {A, B, C, D, E, F}, k = 3. Ánh xạ f có thể cho như sau:

i	1	2	3
f(i)	E	C	E

  - f = (E, C, E) là chỉnh hợp lặp chập 3 của S.
  - **Số chỉnh hợp lặp chập k của tập n phần tử là n^k.**

Một số kiến thức về đại số tổ hợp

- **Chỉnh hợp không lặp:**
  - Khi ánh xạ f là một đơn ánh: ∀i, j ∈ X ta có f(i) = f(j) ⇔ i = j.
  - Hay dãy f(1), f(2), …, f(k) đôi một khác nhau.
  - Ví dụ: S = {A, B, C, D, E, F}, k = 3. Ánh xạ f có thể cho như sau:

i	1	2	3
f(i)	C	A	E

  - f = (C, A, E) là chỉnh hợp không lặp chập 3 của S.
  - **Số chỉnh hợp không lặp chập k của tập n phần tử là:**

$$P_n^k = n(n-1)(n-2)\ldots(n-k+1) = n!/(n-k)!$$

Một số kiến thức về đại số tổ hợp

- **Hoán vị:**
  - Khi k = n. Một chỉnh hợp không lặp chập n của S được gọi là một hoán vị các phần tử của S.
  - Ví dụ: S = {A, B, C, D, E, F}, k = 6. Ánh xạ f có thể cho như sau:

i	1	2	3	4	5	6
f(i)	C	A	E	B	F	D

  - f = (C, A, E, B, F, D) là một hoán vị của S.
  - **Số hoán vị của tập S gồm n phần tử bằng số chỉnh hợp không lặp chập n của S:**

$$A_n^k = n!$$

Một số kiến thức về đại số tổ hợp

- **Tổ hợp:**
  - Một tập con gồm k phần tử của S được gọi là một tổ hợp chập k của S.
  - Ví dụ: S = {A, B, C, D, E, F}, k = 3.
  - f = (A, D, C) là một tổ hợp chập 3 của S.
  - **Số tổ hợp chập k của S là:**

$$C_n^k = n!/[k!(n - k)!]$$

Phương pháp sinh (generation)

- **Phương pháp sinh có thể áp dụng để giải bài toán liệt kê tổ hợp nếu thỏa mãn hai điều kiện:**
  - Có thể xác định được một thứ tự trên tập các cấu hình tổ hợp cần liệt kê.Từ đó có thể biết được cấu hình đầu tiên và cấu hình cuối cùng.
  - Xây dựng được thuật toán từ một cấu hình chưa phải cấu hình cuối, sinh ra được cấu hình kế tiếp nó.

Phương pháp sinh (generation)

- **Phương pháp sinh có thể mô tả như sau:**

  Xây dựng cấu hình đầu tiên;

  do {
  - Đưa ra cấu hình đang có;
  - Từ cấu hình đang có sinh ra cấu hình kế tiếp nếu còn;

  } while (chưa hết cấu hình);

Phương pháp sinh (generation)

- **Thứ tự từ điển: Trên các dãy hữu hạn, người ta xác định một quan hệ thứ tự:**
  - Xét a[1..n] và b[1..n] là hai dãy có độ dài n.
  - Trên phần tử của a và b đã có quan hệ thứ tự "≤". Khi đó a ≤ b nếu như:

    Hoặc a[i] = b[i] với ∀i: 1 ≤ i ≤ n.

    Hoặc tồn tại 1 số nguyên k: 1 ≤ k ≤ n để:

    a[1] = b[1]; a[2] = b[2]; …; a[k] = b[k]; a[k+1] ≤ b[k+1];
  - Quan hệ thứ tự a ≤ b này gọi là thứ tự từ điển trên các dãy độ dài n.
  - Khi a, b có độ dài khác nhau ta thêm các phần tử vào cuối dãy a hoặc b và coi những phần tử này nhỏ hơn tất cả những phần tử khác.
  - Ví dụ: (1, 2, 3, 4) < (5, 6); (a, b, c) < (a, b, c, d);

Phương pháp sinh (generation)

- **Sinh các dãy nhị phân độ dài n:**
  - Dãy nhị phân độ dài n là dãy x[1..n] trong đó x[i] ∈ {0, 1}, ∀i: 1 ≤ i ≤ n.
  - Dãy nhị phân x độ dài n là biểu diễn của một giá trị nguyên p(x) ∈ [0, 2^n – 1].
  - Số các dãy nhị phân độ dài n bằng số các số tự nhiên thuộc đoạn [0, 2^n – 1] và là 2^n dãy.
  - Ví dụ: Khi n = 3, các dãy nhị phân độ dài 3 được liệt kê:

p(x)	0	1	2	3	4	5	6	7
x	000	001	010	011	100	101	110	111

  - Mỗi dãy nhị phân tương ứng là một cấu hình cần liệt kê.

Sinh các dãy nhị phân độ dài n

- **Phân tích:**
  - Cấu hình đầu tiên là 00…0, cấu hình cuối cùng là 11…1.
  - Nếu cấu hình đang có x[1..n] không phải cấu hình cuối thì cấu hình kế tiếp nhận được bằng cách cộng thêm 1 (theo cơ số 2 có nhớ) vào cấu hình đang có.
  - Ví dụ: Khi n = 4, các dãy nhị phân độ dài 3 được liệt kê:

Cấu hình đang có	1010	0111
Cộng thêm 1	+ 1	+ 1
Cấu hình kế tiếp	1011	1000

  - Như vậy kỹ thuật sinh cấu hình kế tiếp là: Với cấu hình đang có, xét từ cuối về đầu, tìm gặp số 0 đầu tiên.
    - Nếu thấy thì thay số 0 đó bằng 1, đặt tất cả các phần tử phía sau nó bằng 0.
    - Không thấy thì đây là cấu hình cuối cùng, toàn số 1.

Liệt kê các tập con k phần tử

- **Yêu cầu:**
  Liệt kê các tập con k phần tử của tập S = {1, 2, …, n} theo thứ tự từ điển.
  - Ví dụ: Khi n = 5, k = 3, ta liệt kê đủ 10 tập con (cấu hình) như sau:

  1. {1, 2, 3}; 2. {1, 2, 4}; 3. {1, 2, 5}; 4. {1, 3, 4}; 5. {1, 3, 5}
  6. {1, 4, 5}; 7. {2, 3, 4}; 8. {2, 3, 5}; 9. {2, 4, 5}; 10. {3, 4, 5}

Liệt kê các tập con k phần tử

- **Phân tích:**
  - Cấu hình khởi tạo (đầu tiên) là {1, 2, …, k}
  - Cấu hình kết thúc là {n – k + 1, n – k + 2, …, n}
  - Biểu diễn mỗi cấu hình là 1 dãy x[1..k], trong đó x[1] < x[2] < … < x[k], nghĩa là x là dãy tăng dần.
  - Giới hạn trên của x[k] là n, của x[k-1] là n – 1, của x[k-2] là n – 2;
  - Tổng quát: Giới hạn trên của x[i] là n – k + i;
  - Giới hạn dưới của x[i] là x[i-1] + 1;
  - Với cấu hình x đang có, x là cấu hình kết thúc nếu mọi x[i] đều đạt giới hạn trên. Quá trình sinh kết thúc.
  - Nếu x chưa là cấu hình kết thúc thì sinh ra cấu hình kế tiếp tăng dần và vừa đủ lớn hơn cấu hình đang có.

Liệt kê các tập con k phần tử

- **Ví dụ: n = 9, k = 6.**
  - Cấu hình đang có x = {1, 2, <u>6, 7, 8, 9</u>}
  - Các phần tử từ x[3] đến x[6] đều đạt giới hạn trên.
  - Khi sinh cấu hình kế tiếp ta không thể tăng 1 trong các phần tử từ x[6] về đến x[3], ta phải tăng x[2] = 2 lên thành x[2] = 3.
  - Ta được cấu hình mới là x = {1, 3, 6, 7, 8, 9}
  - Cấu hình mới này thỏa mãn lớn hơn cấu hình đang có, nhưng chưa thỏa mãn tính vừa đủ nên ta thay các phần tử từ x[3] đến x[6] bằng giới hạn dưới của nó. Tức là:
    x[3] = x[2] + 1 = 4
    x[4] = x[3] + 1 = 5
    x[5] = x[4] + 1 = 6
    x[6] = x[5] + 1 = 7
  - Ta được cấu hình kế tiếp là x = {1, 3, 4, 5, 6, 7}

Liệt kê các tập con k phần tử

- **Kỹ thuật sinh cấu hình kế tiếp như sau:**
  - Tìm từ cuối dãy x về đầu cho tới khi gặp phần tử chưa đạt giới hạn trên n – k + i
  - Nếu tìm thấy, giả sử là x[i]:
    - Tăng phần tử x[i] đó lên 1.
    - Đặt tất cả các phần tử sau x[i] đó bằng giới hạn dưới tương ứng của chúng.
  - Nếu không tìm được, nghĩa là các phần tử đều đạt giới hạn trên, đây là cấu hình kết thúc.

Liệt kê các hoán vị

- **Yêu cầu:**
  Liệt kê các hoán vị của tập S = {1, 2, …, n} theo thứ tự từ điển.
  - Ví dụ: Khi n = 3, ta liệt kê đủ 6 hoán vị (cấu hình) như sau:
    1. {1, 2, 3}; 2. {1, 3, 2}; 3. {2, 1, 3};
    4. {2, 3, 1}; 5. {3, 1, 2}; 6. {3, 2, 1};
  - Cấu hình đầu tiên là x = {1, 2, …, n}
  - Cấu hình kết thúc là x = {n, n-1, …, 1}
  - Hoán vị kế tiếp phải lớn hơn và lớn hơn vừa đủ so với cấu hình đang có.

Liệt kê các hoán vị

- **Phân tích:**
  - Giả sử cấu hình (hoán vị) đang có là x = {3, 2, 6, 5, 4, 1}
  - Xét 4 phần tử cuối cùng x[3] đến x[6], ta thấy chúng được xếp theo thứ tự giảm dần.
  - Nghĩa là ta hoán vị 4 phần tử này theo bất kỳ thứ tự nào cũng đều thu được cấu hình mới bé hơn cấu hình đang có.
  - Như vậy phải xét x[2] = 2; thay nó bằng một giá trị khác;
  - Thay bằng 1, cấu hình kế tiếp sẽ bé hơn.
  - Thay bằng 3, không được vì đang có x[1] = 3 (Phần tử đứng sau không được chọn vào những giá trị của các phần tử đứng trước).
  - Các phần tử còn lại là 6, 5, 4. Vì cấu hình kế tiếp cần vừa đủ lớn nên lấy x[2] = 4.
  - x[3] đến x[6] lấy trong tập {6, 5, 2, 1}. Vì tính chất vừa đủ lớn nên lấy x[3] = 1, x[4] = 2, x[5] = 5, x[6] = 6;

Liệt kê các hoán vị

- **Phân tích:**
  - Vậy cấu hình kế tiếp là x = {3, 4, 1, 2, 5, 6}
- **Nhận xét: Với cấu hình đang có là x = {3, 2, 6, 5, 4, 1}**
  - Đoạn cuối của x được xếp giảm dần.
  - Phần tử x[5] = 4 là số nhỏ nhất ở đoạn cuối, thỏa mãn lớn hơn x[2] = 2.
  - Ta đảo giá trị của x[2] với x[5], được x[2] = 4. Đoạn cuối vẫn được sắp xếp giảm dần.
  - Muốn biểu diễn nhỏ nhất cho đoạn cuối, ta đảo ngược đoạn cuối.
  - Trường hợp cấu hình đang xét là x = {2, 1, 3, 4}, thì cấu hình kế tiếp là x = {2, 1, 4, 3}

Liệt kê các hoán vị

- **Kỹ thuật sinh cấu hình kế tiếp (nếu cấu hình hiện tại chưa là cấu hình kết thúc)**
  - Duyệt dãy x từ x[n-1] về đầu để xác định đoạn cuối giảm dần dài nhất, tìm chỉ số i của phần tử x[i] là phần tử đứng liền trước đoạn cuối đó, x[i] < x[i+1].
  - Nếu tìm được chỉ số i như trên
    - Tìm phần tử nhỏ nhất trong đoạn cuối, giả sử là x[k], thỏa mãn x[k] > x[i] (duyệt đoạn cuối từ x[n] về đến x[i+1]).
    - Đảo giá trị của x[i] với x[k].
    - Đảo ngược đoạn cuối (x[i+1] đến x[n]) để được đoạn cuối xếp tăng dần
  - Nếu không tìm thấy chỉ số i thì cấu hình đang xét đã là cấu hình kết thúc.