

TRẦN ĐỖ HÙNG

THUẬT TOÁN THÔNG DỤNG

TÀI LIỆU BỒI DƯỠNG HỌC SINH GIỎI TIN HỌC
DÀNH CHO HỌC SINH, SINH VIÊN

NHÀ XUẤT BẢN
KHOA HỌC & KỸ THUẬT



TRẦN ĐỖ HÙNG

THUẬT TOÁN THÔNG DỤNG

**TÀI LIỆU BỒI DƯỠNG HỌC SINH GIỎI TIN HỌC
DÀNH CHO HỌC SINH, SINH VIÊN**



**NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
HÀ NỘI - 2010**

Biên mục trên xuất bản phẩm của Thư viện Quốc gia Việt Nam

Trần Đỗ Hùng

Thuật toán thông dụng : Tài liệu bồi dưỡng học sinh giỏi Tin học dành cho học sinh, sinh viên / Trần Đỗ Hùng. - H. : Khoa học và Kỹ thuật, 2010. - 368tr. : hình vẽ : 24cm

Thư mục: tr. 365.

1. Thuật toán 2. Cấu trúc dữ liệu 3. Tin học
005.1 - dc14

KKF0001p-CIP

LỜI NÓI ĐẦU

Thuật toán thông dụng và cấu trúc dữ liệu cơ bản là những kiến thức cơ sở trong lập trình. Đây cũng là những chuyên đề được đưa vào chương trình Tin học của học sinh, sinh viên Chuyên Tin. Đặc biệt, nội dung này cần thiết khi giải các bài toán Tin học dành cho học sinh giỏi Trung học Phổ thông.

Do đã có một số cuốn sách tham khảo trước đây và tài liệu giáo khoa Chuyên Tin đề cập tới các chuyên đề riêng biệt thuộc lĩnh vực này nên trong cuốn "Thuật toán thông dụng" tác giả không đề cập lại chi tiết những kiến thức đó. Tuy nhiên, đôi chỗ do mức độ cần thiết tác giả có sơ lược tóm tắt lại để việc trình bày được hệ thống.

Cuốn "Thuật toán thông dụng" gồm 6 chương:

Chương I. Giới thiệu các kiến thức cơ sở: Một số cấu trúc dữ liệu cơ bản. Dệ quy. Một số cấu trúc dữ liệu đặc biệt.

Chương II. Thuật toán lý thuyết số.

Chương III. Một số thuật toán tìm kiếm.

Chương IV. Một số thuật toán sắp xếp.

Chương V. Nguyên lý tham (Greedy).

Chương VI. Một vài bài tập về lý thuyết trò chơi.

Mỗi chương gồm ba mục: (1) Lý thuyết và các ví dụ cơ bản (2) Bài tập minh họa và hướng dẫn giải. (3) Bài tập tự giải.

Các bài tập minh họa được trình bày bằng ngôn ngữ Pascal hoặc C/C++. Để giúp các em học sinh, sinh viên làm quen với ngôn ngữ C/C++, trong một số bài tập tác giả nêu lời giải bằng cả hai ngôn ngữ để tiện so sánh.

Tác giả chân thành cảm ơn nhà xuất bản Khoa học và Kỹ thuật đã cho xuất bản cuốn sách này và rất mong nhận được những ý kiến đóng góp quý báu của các độc giả.

Tác giả

Chương 1. KIẾN THỨC CƠ SỞ

I. MỘT SỐ CẤU TRÚC DỮ LIỆU CƠ BẢN

1. Mảng

Mảng một chiều là dãy hữu hạn các phần tử cùng kiểu. Các phần tử của mảng được đánh số thứ tự (gọi là chỉ số của phần tử). Một phần tử của mảng được xác định bởi tên mảng và chỉ số của nó. Ví dụ mảng A có các phần tử được đánh số từ 0 đến N ($N \in \mathbb{Z}$), thì phần tử thứ ba của mảng là $A[2]$. Mảng B đánh số theo thứ tự 'a', 'b', 'c', ..., thì phần tử thứ ba là $A['c']$. Các phần tử của mảng được lưu trữ bởi các ô liên tiếp nhau trong bộ nhớ. Có thể truy cập tới một phần tử bất kỳ của mảng bằng cách truy cập trực tiếp vào ô nhớ của phần tử đó trong bộ nhớ, nên mảng có cấu trúc truy cập ngẫu nhiên.

Kích thước của mảng trong bộ nhớ bằng số phần tử của mảng nhân với kích thước của một phần tử trong bộ nhớ.

Mảng hai chiều được coi là mảng một chiều có các phần tử là mảng một chiều cùng kiểu, cùng kích thước. Mỗi phần tử của mảng hai chiều có hai chỉ số: chỉ số dòng và chỉ số cột. Ví dụ mảng hai chiều A có chỉ số dòng và chỉ số cột được đánh số từ 0 đến N thì $A[2, 5]$ là phần tử ở dòng thứ ba và cột thứ sáu (hoặc trong C/C++ kí hiệu là $A[2][5]$).

Sau đây là một ví dụ minh họa sử dụng mảng:

Bài toán 1. (Bài toán $3n+1$). Các bài toán trong Tin học được phân loại thành các lớp tương ứng với thuật toán giải nó. Bài toán này giới thiệu một thuật toán mà đến nay chưa thể phân loại nó, mặc dù có thể tiến hành kiểm nghiệm (test) nó với rất nhiều bộ dữ liệu vào khác nhau:

Bước 1. Nhập số nguyên n

Bước 2. In số n

Bước 3. Nếu $n=1$ thì dừng

Bước 4. Nếu n lẻ thì $n \leftarrow 3n+1$ (gán giá trị $3n+1$ cho n)
 còn không thì $n \leftarrow n/2$ (gán $n/2$ cho n)

Bước 5. Trở về bước 2.

Ví dụ: Cho $n = 22$, thực hiện thuật toán trên thì dãy sau đây được in ra:

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Ta phỏng đoán rằng thuật toán sẽ dừng (khi 1 được in ra) với mọi số nguyên n được đưa vào làm input. Mặc dù thuật toán trông rất đơn giản nhưng ta không chắc rằng phỏng đoán của ta đúng hay không? Chúng ta có thể tiến hành thăm tra phỏng đoán này bằng mọi số nguyên n ($0 \leq n \leq 1000000$). Ta gọi số lượng các số được in ra cho đến khi gặp số 1 (kể cả số 1) là độ dài chu trình của số n . Trong ví dụ trên độ dài chu trình của $n = 22$ là 16. Cho hai số nguyên dương bất kỳ i và j , bạn hãy viết chương trình xác định độ dài chu trình lớn nhất của các số nguyên trong đoạn $[i; j]$.

Input. Tập dữ liệu vào chứa một dãy các cặp số nguyên dương i và j ; mỗi cặp số trên một dòng. Giả sử chỉ xét các số nguyên trong khoảng $(0; 10000)$. Với mỗi cặp i, j hãy xác định độ dài chu trình lớn nhất của các số nguyên trong đoạn $[i; j]$.

Output. Với mỗi cặp số nguyên dương i và j trong tập input, chương trình của bạn cần xuất ra các số i, j và độ dài chu trình lớn nhất của các số nguyên trong đoạn $[i; j]$. Ba số này cách nhau ít nhất một dấu trống và cả ba số trên một dòng: mỗi dòng output tương ứng với một dòng input. Các số nguyên i và j cần xuất hiện trong output như thứ tự xuất hiện trong input, cuối cùng là độ dài chu trình lớn nhất của các số nguyên trong đoạn $[i; j]$. *Ví dụ:*

Input	output
1 10	1 10 20
100 200	100 200 125
201 210	201 210 89
900 1000	900 1000 174

Hướng dẫn. Tạo mảng A một chiều. $A[t]$ là số lượng số cần sinh ra từ số t để gặp lại số 1. Khởi trị $A[i] = -1$ với mọi i (có nghĩa là chưa xét số i), $A[1] = 1$ (khởi trị này là tất nhiên vì số 1 đã có sẵn nên dãy chỉ dài là 1).

Tìm chu trình của số t thuộc đoạn $[i; j]$ bằng cách sau: Thực hiện thuật toán nêu trong đề bài bắt đầu từ t để sinh ra các số mới, đếm số lần thực hiện (tức là số lần sinh ra một số mới, số lần này kí hiệu là nc) cho đến khi gặp một số z đã sinh ra trước đó (thể hiện là $A[z] > 0$). Từ t đến z có nc số được sinh ra và từ z đến 1 có $A[z]$ số sinh ra do đó $A[t] = nc + A[z]$.

Hàm `cycle(i; j)` để tìm chu trình dài nhất trong các chu trình của các số t $[i; j]$.

Chú ý rằng, trong quá trình sinh số mới có thể sinh ra các số lớn hơn 10000, vẫn phải thực hiện quá trình sinh số mới nêu trong đề bài, mặc dù các số cần tìm chu trình chỉ trong phạm vi (0; 10000).

Chương trình Pascal

```
uses crt;
const MAXN=10000;
      fi   = 'bt1.dat';
      fo   = 'bt1.out';
var   A    : array[0..MAXN + 1] of longint;
      i, j, n : longint;
      f, g   : text;
function cycle(i, j : integer) : integer;
var t, max, number, nc : longint;
begin
  max := 0;
  for t:=i to j do
  begin
    nc := 0;
    number := t;
    while ((number>MAXN) or (A[number] < 0)) do
```



```

begin
    nc := nc+1;
    if (number mod 2 = 1) then
        number := 3*number + 1
    else
        number := number div 2;
    end;
    nc := nc + A[number];
    A[t] := nc;
    if (nc > max) then max := nc;
end;
cycle := max;
end;
BEGIN
    assign(f,fi); reset(f);
    assign(g,fo); rewrite(g);
    for i:= 0 to MAXN do A[i] := -1;
    A[1] := 1;
    while not eof(f) do
        begin
            readln(f,i,j);
            writeln(g,i,'j', cycle(i,j) );
        end;
        close(f); close(g);
    END.

```

Chương trình (C++)

```

#include <iostream>
#include <fstream>
const int MAXN=10000;
int A[MAXN + 1];
int cycle(int i, int j) {
    int t, max, number, nc;

```

```

max = 0;
for (t = i; t <= j; t++) {
    nc = 0;
    number = t;
    while ((number > MAXN) || (A[number] < 0)) {
        nc++;
        if (number % 2 == 1)
            number += (number << 1) + 1;
        else
            number = (number >> 1);
    }
    nc += A[number];
    A[t] = nc;
    if (nc > max) max = nc;
}
return max;
}
int main() {
int i,j,n;
ifstream fi("C:\\Baitap\\bt1.dat");
ofstream fo("C:\\Baitap\\bt1.out");
for (i = 0; i <= MAXN; i++) A[i] = -1;
A[1] = 1;
while (fi > 0) {
    fi >> i >> j;
    if (fi == 0) break;
    cout << i << ' ' << j << ' ' << cycle(i,j) << endl;
}
return 0;
}

```