

# Bài toán lập lịch

Nguyễn Thế Anh

## I. Một số thuật toán

### 1. Thuật toán Johnson

*Bài toán 1:*

'Mỗi một chi tiết  $D_1, D_2, \dots, D_n$  cần phải được lần lượt gia công trên 2 máy A, B. Thời gian gia công chi tiết  $D_i$  trên máy A là  $a_i$  trên máy B là  $b_i$  ( $i=1, 2, \dots, n$ ). Hãy tìm lịch (trình tự gia công) các chi tiết trên hai máy sao cho việc hoàn thành gia công tất cả các chi tiết là sớm nhất'.

**Thuật Johnson:**

Chia các chi tiết thành 2 nhóm: nhóm N1, gồm các chi tiết  $D_i$  thoả mãn  $a_i < b_i$ , tức là  $\min(a_i, b_i) = a_i$  và nhóm N2 gồm các chi tiết  $D_i$  thoả mãn  $a_i > b_i$  tức là  $\min(a_i, b_i) = b_i$ . Các chi tiết  $D_i$  thoả mãn  $a_i = b_i$  xếp vào nhóm nào cũng được.

Sắp xếp các chi tiết trong N1 theo chiều tăng của các  $a_i$  và sắp xếp các chi tiết trong N2 theo chiều giảm của các  $b_i$

Nối N2 vào đuôi N1, dãy thu được (đọc từ trái sang phải) sẽ là lịch gia công.

*Bài toán 2:*

'Xét bài toán gia công N chi tiết trên 3 máy theo thứ tự A, B, C với bảng thời gian  $a_i, b_i, c_i, i=1, 2, \dots, n$  thoả mãn:  $\max b_i \leq \min a_i$  hoặc  $\max b_i \leq \min c_i$ '

**Thuật toán:**

Lịch gia công tối ưu trên 3 máy sẽ trùng với lịch gia công tối ưu trên 2 máy: máy thứ nhất với thời gian  $a_i + b_i$  và máy thứ hai với thời gian  $b_i + c_i$ .

### 2. Thuật toán More

*Bài toán:*

'Có n ô tô đưa vào xưởng sửa chữa được đánh số thứ tự  $1, 2, \dots, n$ . Ô tô phải sửa chữa trong thời gian  $t_i$  và thời điểm phải bàn giao là  $d_i$ . Mỗi thời điểm xưởng chỉ sửa chữa một cái, xưởng sửa chữa không ngừng và thời điểm bắt đầu sửa chữa là 0. Hãy đưa ra một thứ tự sửa chữa sao cho số lượng ô tô đúng hạn là lớn nhất.'

Sắp xếp theo thứ tự tăng dần của thời điểm bàn giao

Duyệt từ đầu cho đến khi gặp ô tô quá hạn đầu tiên (Giả sử là ô tô thứ k)

Tìm từ đầu cho đến ô tô thứ k, ô tô nào có  $t_i$  lớn nhất (Giả sử đó là ô tô thứ m). Nếu ô tô này đã được chuyển một lần rồi thì dừng chương trình, còn nếu không thì ta chuyển ô tô này xuống cuối. Rồi trở lại bước 2.

### 3. Các phương pháp khác:

Thông thường thì các bài toán dạng này thường có rất nhiều cách giải. Nhưng thông dụng nhất là thuật toán quy hoạch động và duyệt nhánh cận. Ngoài ra một số còn đưa về đồ thị.

## II. Bài tập

### Bài 1: Lập lịch ưu tiên đúng hạn

*Đề bài:*

Có n công việc đánh số từ 1 đến n và một máy để thực hiện chúng. **Biết:**

- $p_i$  là thời gian cần thiết để hoàn thành công việc i.
- $d_i$  là thời hạn hoàn thành công việc i.

Mỗi công việc cần được thực hiện liên tục từ lúc bắt đầu cho tới khi kết thúc, không cho phép ngắt quãng.

Khoảng thời gian thực hiện hai công việc bất kỳ chỉ được có nhiều nhất 1 điểm chung. Giả sử  $C_1$  là thời điểm hoàn thành trễ hạn, còn nếu  $C_i < d_i$  thì ta nói công việc i được hoàn thành đúng hạn.

**Yêu cầu:** Tìm trình tự thực hiện các công việc sao cho số công việc được hoàn thành đúng hạn là lớn nhất.

**Dữ liệu:** Vào từ file văn bản LICHHD.INP:

- Dòng đầu tiên chứa số nguyên dương n ( $0 < n < 100$ )
- Dòng thứ 2 chứa n số nguyên dương  $p_1, p_2, \dots, p_n$ .
- Dòng thứ 3 chứa n số nguyên dương  $d_1, d_2, \dots, d_n$ .

**Kết quả:** Ghi ra file văn bản LICHHD.OUT.

- Dòng đầu tiên ghi số lượng công việc được hoàn thành đúng hạn theo trình tự thu được.
- Dòng tiếp theo ghi trình tự thực hiện các công việc đã cho.

**Ví dụ :**

*Hướng Dẫn:*

Sắp Xếp theo thứ tự tăng dần của  $D[i]$

*Chúng ta xét như sau:*

Đầu tiên cho:

time:=0;

+ Lấy thực hiện các việc theo trật tự tăng dần. Nhưng đến công việc nào mà Thời gian thực hiện nó sẽ vượt qua giới hạn  $D[i]$  của nó thì:

Tìm các công việc đã được xếp trước nó, công việc nào được thay thế bởi công việc này nếu thời gian thực hiện xong nó là nhỏ nhất.

*Các thủ tục của bài toán:*

Qsort (tăng theo giá trị  $D[i]$ )

Thủ tục Xep(i:Integer) là thủ tục xếp việc:

```
procedure xep(x : integer);
```

```
var
```

```
    i,t,cs : integer;
```

```
    min : integer;
```

```
begin
```

```
    inc(shd);
```

```
    st[shd]:=x;
```

```
    if time+p[x]<=d[x] then
```

```
    begin
```

```
        time:=time+p[x];
```

```
        exit;
```

```
    end;
```

```
    min:=time;
```

```
    cs:=shd;
```

```
    for i:=1 to shd-1 do
```

```
    begin
```

```
        t:=time+p[x]-p[stt[i]];
```

```
        if (t<min) and (t<=d[x]) then
```

```
        begin
```

```
            min:=t;
```

```
            cs:=i;
```

```
        end;
```

```
    end;
```

```
    dec(shd);
```

```
    for i:=cs to shd do stt[i]:=stt[i+1];
```

```
    time:=min;
```

```
end;
```

Trong đó *shd* là số hiệu đỉnh các công việc được thực hiện. Còn mảng *Stt* là mảng nêu tuần tự các công việc được thực hiện.

## **Bài 2:**

*Đề bài:* Bài toán gia công trên 2 máy

*Hướng dẫn:*

Làm theo thuật toán Johnson

```
{ $A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+,Y+ }
```

```
{ $M 16384,0,655360 }
```

```
program Gia_cong_2_may;
```

```
uses crt;
```

```
const
```

```
    fi='giacong.inp';
```

```
    fo='giacong.out';
```

```
var
```

```
    f:text;
```

```

n,sb:integer;
a,b:array[1..100] of integer;
tt:array[1..100] of integer;
roi:array[1..100] of boolean;

procedure input;
var
    i:integer;
begin
    assign(f,fi);
    reset(f);
    readln(f,n);
    for i:=1 to n do readln(f,a[i],b[i]);
    close(f);
end;

procedure khoitao;
begin
    fillchar(tt,sizeof(tt),0);
    fillchar(roi,sizeof(roi),false);
end;

procedure xuly;
var
    j,min,cs,t,k,i:integer;
    ok1,ok2:boolean;
begin
    ok1:=false;
    ok2:=false;
    k:=0;
    t:=0;
    for j:=1 to n do
        begin
            min:=maxint;
            for i:=1 to n do
                begin
                    if (min>a[i]) and (roi[i]=false) then
                        begin
                            cs:=i;
                            min:=a[i];
                            ok2:=false;
                            ok1:=true;
                        end;
                    if (min>b[i]) and (roi[i]=false) then
                        begin
                            cs:=i;
                            min:=b[i];
                            ok1:=false;
                            ok2:=true;
                        end;
                end;
            if ok1=true then
                begin
                    roi[cs]:=true;
                end;
        end;
    end;

```

```

        k:=k+1;
        tt[k]:=cs;
    end
    else
    begin
        tt[n-t]:=cs;
        t:=t+1;
    end;
end;
end;

procedure calc;
var
    sa,i:integer;
begin
    sa:=a[tt[1]];
    sb:=a[tt[1]]+b[tt[1]];
    for i:=2 to n do
    begin
        sa:=sa+a[tt[i]];
        if sa>=sb then sb:=sa+b[tt[i]]
        else sb:=sb+b[tt[i]];
    end;
end;

procedure output;
var
    i:integer;
begin
    assign(f,fo);
    rewrite(f);
    writeln(f,sb);
    for i:=1 to n do write(f,tt[i], ' ');
    close(f);
end;

BEGIN
    clrscr;
    input;
    khoitao;
    xuly;
    calc;
    output;
END.

```

**END.**

Bài toán lập lịch là 1 bài toán khó, trên đây mình chỉ giới thiệu với các bạn 2 thuật toán cơ bản của bài toán lập lịch. Nếu bạn nào quan tâm thì cứ gửi mail cho mình, mình có rất nhiều bài tập nữa liên quan đến phần này.