

BÀI TẬP LẬP LỊCH

Bài 1: Chơng trình máy tính.

Giả sử trong một phiên làm việc từ thời điểm 0 đến thời điểm $T=8640000$, một trung tâm tính toán phải thực hiện N chơng trình, chơng trình i thực hiện từ thời điểm $A[i]$ đến thời điểm $B[i]$, $0 \leq A[i] \leq B[i] \leq T$.

Cho trớc một đoạn thời gian $(P1, Q1)$. Hãy xét xem liệu tại mọi thời điểm của đoạn đó luôn có chơng trình chạy hay không?

Cho trớc một đoạn thời gian $(R1, S1)$. Hãy xét xem liệu tại mọi thời điểm của đoạn đó luôn không có chơng trình nào chạy không?

Hãy tìm đoạn thời gian (P, Q) dài nhất sao cho tại mọi điểm của nó luôn có chơng trình chạy.

Hãy tìm đoạn thời gian (R, S) dài nhất sao cho tại mọi thời điểm của nó đều không có chơng trình nào chạy.

Dữ liệu vào đọc cho bởi file CTMT.INP: dòng đầu ghi số nguyên dương $N \leq 200$. Dòng thứ $i+1$ ($1 \leq i \leq N$) ghi hai số nguyên không âm $A[i]$ và $B[i]$. Dòng thứ $N+2$ ghi hai số nguyên không âm $P1, Q1$ ($P1 \leq Q1$). Dòng thứ $N+3$ ghi hai số nguyên không âm $R1, S1$ ($R1 \leq S1$).

Kết quả ghi ra file CTMT.OUT: dòng thứ nhất ghi 1 hoặc 0 tùy thuộc kết quả cụ thể (tìm được thì ghi số 1, không tìm được thì ghi số 0). Dòng thứ hai cũng ghi số 1 hoặc 0 theo ý nghĩa trên. Dòng thứ ba ghi hai số P, Q . Dòng thứ t ghi hai số R, S .

Ví dụ:

| CTMT.INP | CTMT.OUT |
|-----------------|-----------------|
| 5 | 1 |
| 1000 10000 | 0 |
| 2000 30000 | 8000000 8500000 |
| 20000 100000 | 500001 7999999 |
| 200000 500000 | |
| 8000000 8500000 | |
| 1000 100000 | |
| 0 1000 | |

Bài 2: Xếp việc.

Cho N ($2 \leq N \leq 200$) công việc, mỗi công việc i phải làm trớc một số công việc nào đó trong N công việc này. Hãy xếp lịch thực hiện đủ N công việc.

Dữ liệu vào file XEPVIEC.INP: Dòng đầu là số nguyên dương N . Các dòng tiếp theo thể hiện quan hệ thứ tự bộ phận: đầu dòng là số i , các số tiếp theo là $j_{i1}, j_{i2}, \dots, j_{in}$ thể hiện công việc i phải làm trớc các công việc $j_{i1}, j_{i2}, \dots, j_{in}$.

Kết quả file XEPVIEC.OUT: Một dòng N số hiệu các công việc đọc lần lượt thực hiện.

Ví dụ:

| XEPVIEC.INP | XEPVIEC.OUT |
|-------------|----------------------|
| 10 | 1 2 7 9 4 6 3 5 8 10 |
| 1 2 3 | |
| 2 4 10 | |
| 3 5 | |
| 4 6 8 | |
| 5 8 | |
| 6 3 | |
| 7 9 5 | |
| 9 4 10 | |

Bài 3: Lập lịch gia công trên hai máy.

Một sản phẩm gồm N chi tiết, mỗi chi tiết phải gia công lần lượt trên hai máy A và B (A trớc, B sau). Thời gian thực hiện chi tiết i trên máy A là A_i , trên máy B là B_i ($i=1,2,\dots,N$). Hãy xếp lịch hoàn thành sản phẩm với thời gian ít nhất. In ra lịch và thời gian hoàn thành.

Dữ liệu vào file LLGC2M.INP: Dòng đầu là số nguyên dương N ; N dòng tiếp theo: dòng $i+1$ là hai số A_i và B_i .

Kết quả ghi ra file LLGC2M.OUT: Dòng đầu là thời gian ít nhất thực hiện N công việc. Dòng thứ hai lần lượt ghi số hiệu của N công việc thực hiện theo lịch tối ưu.

Ví dụ:

| LLGC2M.INP | LLGC2M.OUT |
|------------|------------|
| 5 | 26 |
| 3 3 | 1 4 2 5 3 |
| 4 3 | |
| 6 2 | |
| 5 7 | |
| 6 3 | |

Bài 4: Lập lịch chia N việc cho M máy.

Cho N công việc, công việc i hoàn thành trong thời gian t_i . Các công việc được thực hiện trên M máy (công suất nh nhau, mỗi máy đều có thể thực hiện được công việc trong N công việc) mỗi công việc được làm liên tục trên một máy cho đến khi xong. Hãy tổ chức máy thực hiện đủ N công việc sao cho thời gian hoàn thành càng nhỏ càng tốt.

Dữ liệu vào file LLGCMM.INP: Dòng đầu là hai số N và M ; Dòng tiếp theo là N số t_1, t_2, \dots, t_N .

Kết quả ra file LLGCMM.OUT: Dòng đầu là thời gian hoàn thành N công việc; M dòng tiếp: dòng $i+1$ ghi số hiệu các công việc thực hiện trên máy i .

| LLGCMM.INP | LLGCMM.OUT |
|-------------|------------|
| 6 3 | 8 |
| 2 5 8 1 5 1 | 3 |
| | 2 1 4 |
| | 5 6 |

Ví dụ:

Bài 5: Lập lịch thực hiện công việc.

Cho N công việc, với mỗi công việc cho biết tiền công thu được khi thực hiện công việc này, thời gian để hoàn thành, thời điểm cuối cùng phải kết thúc. Hãy xếp lịch thực hiện sao cho thu được nhiều tiền công nhất.

Dữ liệu vào file LLLV.INP: Dòng đầu là số nguyên N ; N dòng sau, dòng thứ $i+1$ ghi 3 số t_i , td_i , gt_i tương ứng là thời gian cần thiết để hoàn thành, thời điểm bắt buộc phải xong, giá trị tiền công của công việc i .

Kết quả ra file LLLV.OUT: Dòng đầu là tổng giá trị tiền công; Các dòng tiếp theo mỗi dòng ghi 4 số: i , T_i , T_2 , gt_i tương ứng là số hiệu, thời điểm bắt đầu, thời điểm kết thúc, giá trị của công việc được chọn.

Ví dụ:

| LLL.V.INP | LLL.V.OUT |
|-----------|------------|
| 10 | 329 |
| 1 4 89 | 5 0 1 25 |
| 5 5 86 | 1 1 2 89 |
| 4 11 83 | 3 2 6 83 |
| 5 7 84 | 6 6 9 61 |
| 1 2 25 | 10 9 14 71 |
| 3 11 61 | |
| 6 11 33 | |
| 4 7 28 | |
| 3 10 1 | |
| 5 14 71 | |

Bài 6: Xếp lịch học tập:

Một anh sinh viên khoa công nghệ thông tin (do mãi chơi lên lập trình cha tốt) đang băn khoăn không biết phải phân phối thời gian học N ($N \leq 100$) môn nh thế nào trong một khoảng thời gian còn lại là T . Mỗi môn học i có số học trình là S_i . (Khi tính điểm tổng kết, điểm thi một môn được nhân với số học trình S_i của môn đó vì số học trình càng cao thì lượng kiến thức đào tạo của môn đó càng nhiều).

Kết quả của môn học tùy thuộc vào lượng thời gian anh ta dành cho môn đó. Anh ta không muốn thi lại bất cứ môn nào và muốn có điểm tổng kết cao nhất.

Hãy lập trình giúp anh sinh viên đó phân phối thời gian học các môn.

Dữ liệu vào file Hoctrinh.inp: Dòng đầu là hai số N và T ; Dòng thứ hai là N số S_i ($1 \leq i \leq N$); N dòng tiếp theo: dòng thứ i trong N dòng này chứa 6 số thể hiện số giờ tối thiểu để anh ta đạt được điểm lần lượt là 5 đến 10.

Kết quả ra file Hoctrinh.out: Dòng đầu là tổng điểm của N môn (đã tính hệ số học trình); N dòng tiếp theo, dòng thứ i là thời gian mà anh ta đầu tư cho môn thứ i .

Ví dụ:

| Hoctrinh.inp | Hoctrinh.out |
|--------------|--------------|
| 3 12 | 43 |
| 1 2 2 | 1 |
| 1 2 3 4 5 6 | 5 |
| 1 2 3 4 5 6 | 6 |
| 1 2 3 4 5 6 | |

Bài 7: Điều động nhân viên tiếp thị.

Một công ty hoạt động kinh doanh trên N vùng lãnh thổ đánh số từ 1 đến N . Biết chi phí đi lại giữa thành phố i và thành phố j là $C[i,j]$ ($C[i,j] \leq 300$). Công ty cần điều động K nhân viên tiếp thị đang hoạt động ở K thành phố khác nhau trong N thành phố nói trên, mỗi nhân viên về công tác ở một trong K địa điểm mới này ($2K < N < 100$).

Hãy tìm cách điều động các nhân viên tiếp thị đến địa điểm hoạt động mới sao cho tổng chi phí cho việc di chuyển chỗ làm việc của các nhân viên này là nhỏ nhất có thể.

Dữ liệu vào file DDNV.INP: Dòng đầu là hai số N và K cách nhau một dấu cách; N dòng tiếp theo, mỗi dòng chứa N số nguyên, dòng thứ i chứa chi phí đi lại từ thành phố i đến các thành phố còn lại, quy ước $C[i,i]=0$; Dòng tiếp theo chứa K số nguyên tương ứng là chỉ số các địa điểm đang hoạt động của K nhân viên tiếp thị; Dòng cuối cùng chứa K số nguyên tương ứng là chỉ số các địa điểm nơi cần điều động K nhân viên tiếp thị nói trên đến hoạt động.

Kết quả ra file DDNV.OUT: Dòng đầu chứa tổng chi phí theo cách điều động tìm được; Dòng thứ i trong số K dòng tiếp theo ghi cách di chuyển của nhân viên thứ i bắt đầu từ thành phố đang hoạt động đến địa điểm hoạt động mới của mình (có thể phải di chuyển qua một số thành phố trung gian) dưới dạng dãy có thứ tự các thành phố phải đi qua.

| DDNV.INP | DDNV.OUT |
|------------------|----------|
| 6 2 | 60 |
| 0 20 60 30 50 70 | 1 4 |
| 30 0 70 50 20 30 | 3 5 6 |
| 40 70 0 80 20 50 | |
| 70 60 80 0 30 20 | |
| 50 80 30 40 0 10 | |
| 50 40 70 90 20 0 | |
| 1 3 | |
| 4 6 | |

Ví dụ:

Bài 8: Nói dây liên lạc.

Trong hội trợ triển lãm người ta cần thiết lập mạng liên lạc nội bộ cho tất cả các gian hàng triển lãm. Cho biết khoảng cách của các gian hàng hiện tại được biểu diễn bằng ma trận $A[i,j]$ trong đó $a[i,j]=a[j,i]$ là chiều dài đồng dây nối từ gian hàng i sang gian hàng j ($a[i,j]=a[j,i]=0$ nếu không cần nối từ i sang j).

Yêu cầu: Hãy viết chương trình tính độ dài dây dẫn ngắn nhất có thể sao cho mọi gian hàng đều có thể liên lạc được với nhau trực tiếp hoặc thông qua các gian hàng trung gian khác.

Tên file chương trình: **Noiday.pas**.

Dữ liệu vào file **Noiday.inp**: Dòng đầu là N ($N \leq 100$) số gian hàng và N dòng tiếp theo là ma trận biểu diễn khoảng cách giữa các gian hàng cần hay không cần nối dây giữa các gian hàng.

Kết quả ra file **Noiday.out**: Dòng đầu là tổng chiều dài dây nối; Các dòng tiếp theo là phương án nối dây giữa các gian hàng.

Ví dụ:

| Noiday.inp | Noiday.out |
|------------|------------|
| 4 | 4 |
| 0 1 5 2 | 3->4 |
| 1 0 3 2 | 1->2 |
| 5 3 0 1 | 1->4 |
| 2 2 1 0 | |
