TRẠI HÈ HÙNG VƯƠNG LẦN THỨ X
TRƯỜNG THPT CHUYÊN HOÀNG VĂN THỤ
TỈNH HÒA BÌNH

ĐỀ THI ĐỀ XUẤT

ĐỀ THI MÔN TIN HỌC
KHỐI 11
(Đề này có 3 trang, gồm 3 bài)

Tổng quan đề thi

Tên bài	Tên file	INPUT	OUTPUT	Thời gian	Điểm
Giải phương trình	GIAIPT.PAS	GIAIPT.INP	GIAIPT.OUT	1 s	7
Ma trận con	MATRIX.PAS	MATRIX.INP	MATRIX.OUT	1 s	7
Trọng tải xe	WEIGHT.PAS	WEIGHT.INP	WEIGHT.OUT	1 s	6

Bài 1.

Cho trước phương trình:

$$x^2 + S(x) * x - N = 0$$

Trong đó x, N là những số nguyên dương, S(x) bằng tổng các chữ số của x.

Bạn được cho trước giá trị N, hãy tìm giá trị x nhỏ nhất thỏa mãn phương trình trên.

INPUT: GIAIPT.INP

- Một số nguyên duy nhất N ($1 \le N \le 10^{18}$)

OUTPUT: GIAIPT.OUT

- Một số nguyên duy nhất x nhỏ nhất thỏa mãn phương trình. Trong trường hợp không tìm được x thì hiện ra -1.

Ví dụ:

GIAIPT.INP	GIAIPT.OUT
2	1
4	-1

* Chú ý:

+ 50% test $N \le 10^4$ và có nghiệm $x \le 10^4$
+ 70% test $N \le 10^9$ và có nghiệm $x \le 10^5$
+ 100% test $N \le 10^{18}$

Câu 2.

Cho ma trận A kích thước M x N, trong đó các dòng được đánh thứ tự từ 1 đến M trừ trên xuống, các cột được đánh thứ tự từ 1 đến N từ trái sang phải.

Tại mỗi ô của bảng có chứa giá trị nguyên A[i,j].

Nhiệm vụ của bạn là tìm ma trận con có tổng các phần tử trong ma trận đó là lớn nhất.

INPUT: MATRIX.INP

- Dòng đầu tiên là hai số nguyên M và N ($1 \le M, N \le 500$)
- M dòng tiếp theo, dòng thứ i chứa N số $A_{i1}, A_{i2}, \ldots, A_{iN}$ ($|A_{ij}| \le 5*10^4$)

OUTPUT: MATRIX.OUT

- Một dòng duy nhất là tổng lớn nhất của các phần tử thuộc ma trận con tìm được.

Ví dụ:

MATRIX.INP	MATRIX.OUT
3 5 -4 5 -18 9 5 -16 4 0 -4 9 5 -1 4 -1 2	20

* Giải thích: ma trận con có tổng lớn nhất từ vị trí (dòng 1, cột 4) đến vị trí (dòng 3, cột 5)

* Chú ý:

+ 50% test $N \le 10$
+ 70% test $N \le 100$
+ 100% test $N \le 500$

Câu 3.

Có N thành phố và M đường đi vô hướng nối giữa một số cặp thành phố. Mỗi đường đi u, v có một giới hạn về trọng tải là l và r ($l \le r$) (có thể có nhiều đường đi trực tiếp từ u đến v), tức là muốn di chuyển từ thành phố u sang thành phố v thì ô tô phải có trọng tải x ($l \le x \le r$).

HN đang có ý định mua cho mình 1 chiếc ô tô để đi làm. Nhà của HN ở thành phố 1, cơ quan ở thành phố N. Hãy giúp HN xem có bao nhiêu cách lựa chọn mua một chiếc ô tô có trọng tải x để HN hàng ngày có thể đi làm từ thành phố 1 đến thành phố N.

INPUT: WEIGHT.INP

- Dòng 1 chứa hai số nguyên N và M ($1 \le N \le 10^3$, $0 \le M \le 5*10^3$)
- M dòng tiếp theo, dòng thứ i gồm 4 số u, v, l, r ($1 \le u, v \le n$, $1 \le l \le r \le N$) biểu diễn 1 con đường nối 2 thành phố u, v có giới hạn trọng tải từ l tới r.

OUTPUT: WEIGHT.OUT

- Một giá trị K duy nhất là số cách lựa chọn ô tô có trọng tải x để có thể đi từ thành phố 1 tới thành phố N.
- Trong trường hợp không có cách chọn xe thì ghi ra -1

Ví dụ:

WEIGHT.INP	WEIGHT.OUT
4 4 1 2 1 10 2 4 3 5 1 3 1 5 3 4 2 7	6

* Giải thích: có 6 lựa chọn xe ô tô có trọng tải từ 2 đến 7.

* Chú ý:

+ 60% test mỗi cặp (u,v) chỉ có duy nhất 1 đường nối trực tiếp
+ 40% test còn lại mỗi cặp (u,v) có thể có nhiều đường nối trực tiếp

.................HẾT.....................

Người ra đề

Vương Thành Trung – đt 0984800929

Hướng dẫn thuật toán

Câu 1.

+ 50% test đầu tiên $N \le 10^4$ và có nghiệm $x \le 10^4$

+ 20% test tiếp theo $N \le 10^9$ và có nghiệm $x \le 10^5$

+ 30% test cuối cùng $N \le 10^{18}$

* Với 2 subtest đầu tiên dễ dàng thấy ngay chỉ cần duyệt x từ 1 đến 10^4 (subtest 1) và 10^5 (subtest 2). Trong quá trình duyệt, nếu có x thỏa mãn phương trình thì x là nghiệm nhỏ nhất tìm được.

- Nếu HS khai báo x: integer thì chạy được Subtest1
- Nếu HS khai báo x: longint thì chạy được subtest2

Độ phức tạp $O(10^5)$

* Ở subtest 3 đề bài không giới hạn x nhưng có thể suy ra giới hạn x như sau:

Từ phương trình: $x^2 + S(x)*x - N = 0 \Rightarrow x^2 \le N \le 10^{18} \Rightarrow x \le 10^9 \Rightarrow 1 \le S(x) \le 90$

Khi đó duyệt S(x) từ 1 đến 90, và giải phương trình tìm x nhỏ nhất.

Độ phức tạp O(90)

Câu 2.

+ 50% test $N \le 10$

+ 70% test $N \le 100$

+ 100% test $N \le 500$

* Subtest1:

Duyệt 2 góc trái trên và phải dưới của ma trận ta được các ma trận con, với mỗi ma trận con duyệt tiếp để tính tổng các phần trong ma trận này, cập nhật tổng lớn nhất

Độ phức tạp $O(M^3*N^3)$

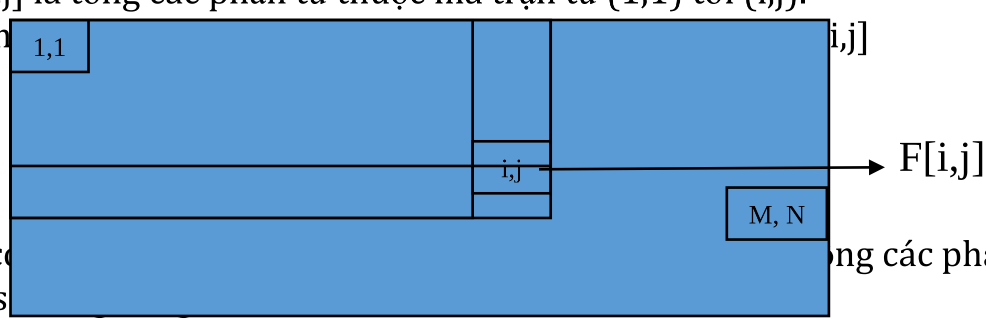
* Subtest2:

Sử dụng ý tưởng QHĐ:

Gọi F[i,j] là tổng các phần tử thuộc ma trận từ (1,1) tới (i,j).

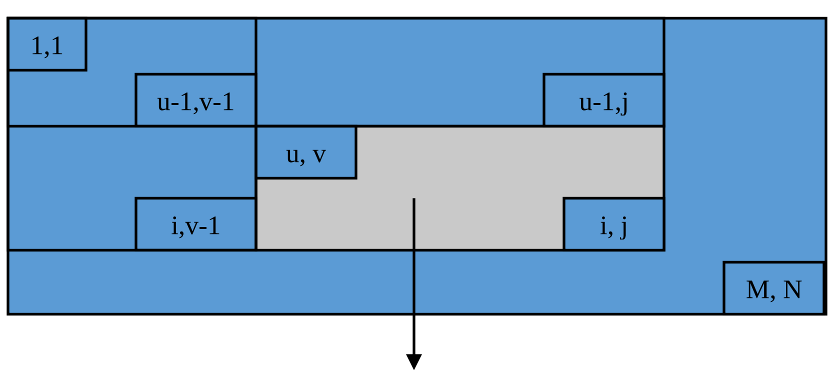
Công th i,j]

Ma trận F



Xét ma trận c ng các phần tử trong ma trận này s

$$S := F[i,j] + F[u-1,v-1] - F[u-1,j] - F[i,v-1]$$



Ma trận con cần tính tổng các phần tử

Độ phức tạp $O(M^2*N^2)$

* Subtest 3:

Trước hết cần giải bài toán con sau:

Cho dãy số $A_1, A_2, ..., A_N$ tìm đoạn con liên tiếp có tổng lớn nhất.

Cần giải bài toán này với độ phức tạp là O(N)

Gọi S[i] = A1 + A2 + … + Ai

Tìm cặp (i,j) (i ≤ j) sao cho A[i] + A[i+1] + … + A[j-1] + A[j] = S[j] – S[i-1] lớn nhất.

Để S[j] – S[i-1] lớn nhất thì S[i-1] cần có giá trị nhỏ nhất. Từ đó sẽ duyệt j và tìm S[i-1] (với i ≤ j) nhỏ nhất.

Để tìm S[i-1] nhỏ nhất với tốc độ nhanh ta cần chuẩn bị một mảng csmin[i] là chỉ số mà tại đó có S nhỏ nhất (tức là S[csmin[i]] nhỏ nhất)

Tìm csmin[i] như sau:

```
S[0] := 0; csmin[0] := 0
For i := 1 to N do
Begin
        S[i] := S[i-1] + A[i];
        If S[i] < S[csmin[i-1]] then csmin[i] := i
        Else csmin[i] := csmin[i-1];
End;
Tìm Smax như sau:
Smax := -vô cùng
For i:=1 to N do
If i > csmin[i] then
Begin
        Sum := S[i] – S[csmin[i]];
        If Sum > Smax then Smax := Sum
End;
```

Sử dụng bài toán phụ trên để giải bài toán như sau:

Tạo mảng F[i,j] là tổng của từng cột (1,j) tới (i,j)

F[i,j] := F[i-1,j] + A[i,j]

Xét hai dòng i và dòng j:

Tổng dòng i + 1 đến dòng j tại cột k là A[i+1,k] + A[i+2,k] + … + A[j,k] sẽ lưu lại và mảng B[k] (1 ≤ k ≤ N) được tính:

B[k] := F[j,k] – F[i-1,k]

Khi đó ta sẽ tìm tổng lớn nhất của các phần tử liên tiếp trên mảng B, cập nhật tổng lớn và đó là giá trị cần tìm. Áp dụng bài toán phụ ở trên.

Độ phức tạp: $O(M^2*N)$

Câu 3.

+ 60% test mỗi cặp (u,v) chỉ có duy nhất 1 đường nối trực tiếp

+ 40% test còn lại mỗi cặp (u,v) có thể có nhiều đường nối trực tiếp

Sử dụng thuật toán tìm kiếm nhị phân để tìm kết quả bài toán.

Kết quả có thể nằm trong khoảng [1,5000], đặt *dau = 1, cuoi = 5000*, chặt nhị phân kết quả x, với mỗi kết quả x kiểm tra xem trọng tải khoảng [*dau, dau + x – 1*] có thể đi từ 1 -> N không?
Sử dụng DFS hoặc BFS để tìm đường đi từ 1 -> N, điều kiện để có thể đi qua cạnh (u,v) có giới hạn [L, R] là *L <= dau* và *dau + x – 1 <= R.*
Nếu có thể đi từ 1 đến N thì lưu lại kq = x đồng thời tìm cách tăng giá trị x bằng cách gán dau := x + 1, ngược lại nếu không thể đi từ 1 đến N thì giảm x bằng cách gán cuoi := x – 1.
* Subtest 1: có thể sử dụng ma trận kề để lưu đồ thị
* Subtest 2: Sử dụng danh sách kề để lưu đồ thị.