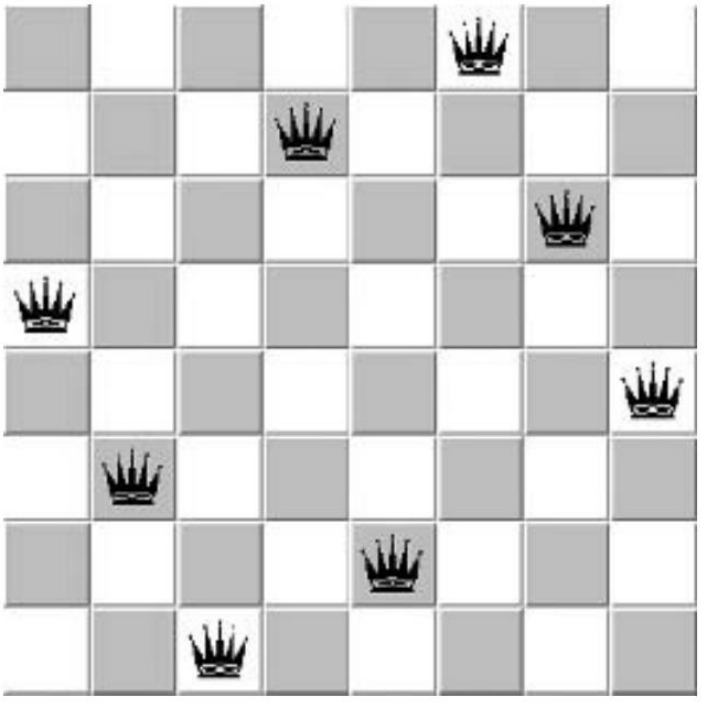
BÀI TOÁN TÁM HẬU

Bài toán tám quân hậu là bài toán đặt tám quân hậu trên bàn cờ vua kích thước 8×8 sao cho không có quân hậu nào có thể "ăn" được quân hậu khác, hay nói khác đi không quân hậu nào có để di chuyển theo quy tắc cờ vua. Màu của các quân hậu không có ý nghĩa trong bài toán này. Như vậy, lời giải của bài toán là một cách xếp tám quân hậu trên bàn cờ sao cho không có hai quân nào đứng trên cùng hàng, hoặc cùng cột hoặc cùng đường chéo. Bài toán tám quân hậu có thể tổng quát hóa thành bài toán đặt n quân hậu trên bàn cờ $n \times n$ ($n \geq 4$).



Dữ liệu: vào từ tập tin văn bản **QUEENS.INP** số nguyên ($4 \leq n \leq 15$) là số quân hậu .

Kết quả: xuất ra tập tin văn bản **QUEENS.OUT**

Một số nguyên là số cách xếp hậu tìm được.

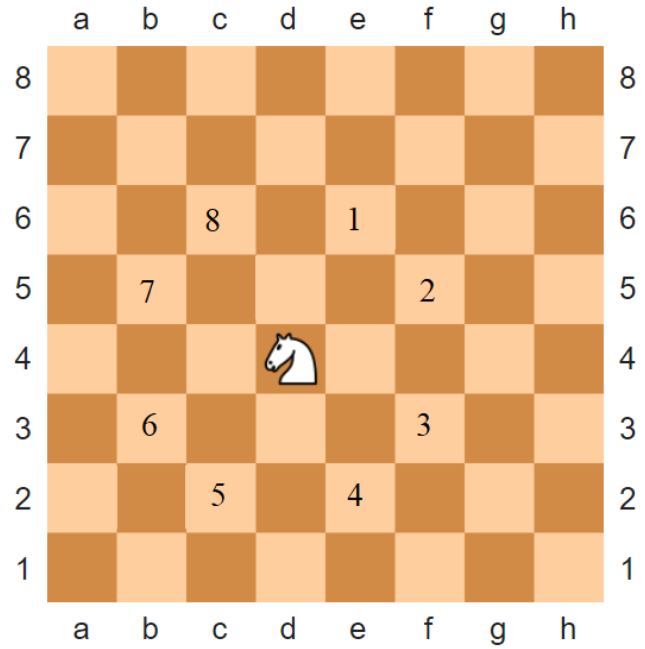
Ví dụ:

QUEENS.INP
8

QUEENS.OUT
92

BÀI TOÁN MÃ ĐI TUẦN

Mã đi tuần hay hành trình của quân mã là bài toán về việc di chuyển một quân mã trên bàn cờ vua (8 x 8). Quân mã được đặt ở một ô trên một bàn cờ trống nó phải di chuyển theo quy tắc của cờ vua để đi qua mỗi ô trên bàn cờ đúng một lần. Quân mã có thể kết thúc tại chính ô mà nó khởi đầu. Một hành trình như vậy được gọi là hành trình đóng. Có những hành trình, trong đó quân mã sau khi đi hết tất cả 64 ô của bàn cờ (kể cả ô xuất phát), thì từ ô cuối của hành trình không thể đi về ô xuất phát chỉ bằng một nước đi. Những hành trình như vậy được gọi là hành trình mở.



Yêu cầu: Nhập vào hai số N và M là kích thước bàn cờ NxM. Hãy cho biết số phương án mã đi tuần theo hành trình đóng của bảng.

Dữ liệu: vào từ tập tin văn bản **KNIGHT.INP** hai số nguyên dương N, M ($5 \le N,M \le 8$) là kích thước của bàn cờ .

Kết quả: xuất ra tập tin văn bản **KNIGHT.OUT**

Một số nguyên dương là số phương án mã đi tuần theo hành trình đóng của bảng.

Ví dụ:

KNIGHT.INP	KNIGHT.OUT
5 5	1728

ĐƯỜNG ĐI TRONG MÊ CUNG

Một mê cung gồm M × N ô vuông (M dòng, N cột, M, N ≤ 100). Mỗi ô vuông có thể có từ 0 đến 4 bức tường bao quanh. Cần phải đi từ bên ngoài vào mê cung, bắt đầu từ phía Tây, qua các ô của mê cung và thoát khỏi mê cung về phía Đông. Chỉ được phép di chuyển theo 4 hướng Đông, Tây, Nam, Bắc và đi qua những nơi không có tường chắn.

1. Trong trường hợp có đường đi, hãy tìm đường đi qua ít ô nhất.

2. Trong trường hợp trái lại, hãy tìm cách phá bỏ ít nhất một số bức tường để có đường đi. Nếu có nhiều phương án như vậy, hãy chỉ ra một phương án để đường đi qua ít ô nhất.

Trạng thái của một ô được cho bởi một số nguyên trong khoảng từ 0 đến 15 theo quy tắc: bắt đầu là 0 (không có tường), cộng thêm 1 (nếu có bức tường phía Tây), cộng thêm 2 (nếu có bức tường phía Bắc), cộng thêm 4 (nếu có bức tường phía Đông), cộng thêm 8 (nếu có bức tường phía Nam).

Dữ liệu: Vào từ file văn bản MECUNG.INP bao gồm

- Dòng đầu ghi giá trị M (số dòng) và N (số cột),
- Các dòng tiếp theo ghi ma trận trạng thái của các ô gồm M dòng, N cột, trong đó giá trị ở dòng i cột j mô tả trạng thái của ô [i, j].

Các giá trị trên cùng một dòng ghi cách nhau ít nhất một dấu trắng.

Kết quả: Ghi ra file văn bản MECUNG.OUT

1. Trường hợp tìm thấy đường đi:

- Dòng đầu ghi số ô mà đường đi đi qua,
- Dòng tiếp theo ghi thông tin về đường đi gồm toạ độ dòng xuất phát, sau đó cách một dấu trắng, là xâu ký tự mô tả đường đi (theo hướng) viết liên tiếp nhau, gồm các ký tự D (đông), B (bắc), T (tây), N (nam). Thí dụ *3 DBBDDNTNDD* mô tả đường đi xuất phát từ dòng 3 sau đó lần lượt đi theo các hướng đông, bắc, bắc, đông, đông, nam, tây, nam, đông, đông.

2. Trường hợp không tìm thấy đường đi:

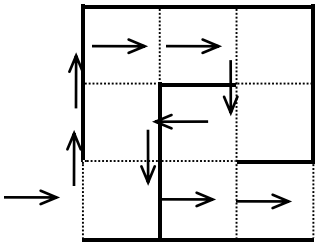
- Dòng đầu ghi số 0,
- Dòng thứ hai ghi số bức tường phải phá,
- Dòng thứ ba ghi số ô mà đường đi đi qua,
- Dòng cuối ghi thông tin về đường đi theo quy cách giống như trường hợp 1.

Ví dụ 1:

MECUNG.INP
3 3 3 10 6 5 3 12 12 9 10

MECUNG.OUT
9 3 DBBDDNTNDD

Hình 1 mô tả một mê cung cho trong ví dụ 1



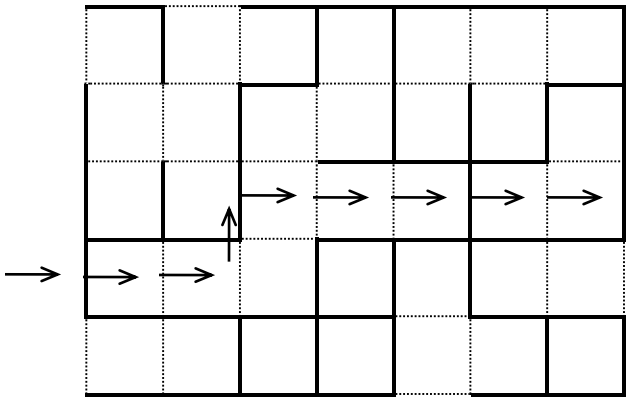
Hình 1

Ví dụ 2:

MECUNG.INP
5 7 6 1 14 7 3 2 14 1 4 11 12 13 13 7 13 13 3 10 14 11 12 11 10 12 15 7 11 10 10 14 15 15 1 14 15

MECUNG.OUT
0 *3* *8* *4 DDDBDDDDD*

Hình 2 mô tả mê cung cho trong ví dụ 2.



Hình 2

Trong ví dụ 2, phải phá ít nhất 3 bức tường để có đường đi. Có nhiều phương án phá 3 bức tường. Phương án nêu trong kết quả có đường đi qua 8 ô (xem hình vẽ) là phương án đi qua ít ô nhất.

ĐƯỜNG ĐI TRÊN LƯỚI

Một mạng đường giao thông gồm n×m nút giao thông có dạng lưới ô vuông gồm các đường phố một chiều ngang dọc nối các nút giao thông. Toạ độ của các nút được đánh số từ trên xuống dưới và từ trái sang phải. Khoảng cách giữa hai nút bất kỳ là như nhau. Mỗi nút giao thông (i, j) nằm ở cao độ a_{ij} mét so với mực nước biển (i=1,...,n; j=1,...,n). Cần tìm đường đi ngắn nhất nối đi từ nút (u_s, v_s) đến nút (u_t, v_t) sao cho trong quá trình di chuyển không khi nào phải di chuyển theo phố mà nút đầu có độ cao thấp hơn p (mét) so với độ cao của nút cuối.

Dữ liệu: Vào từ file văn bản PATH.INP có cấu trúc như sau:

- Dòng đầu tiên ghi ba số n, m, p; (0< n, m <51);
- n dòng tiếp theo mỗi dòng gồm m số nguyên ghi cao độ của các nút giao thông: a_{ij}, j=1, 2,..., m; i=1, 2,..., n;
- Các dòng tiếp theo, mỗi dòng gồm 4 số nguyên dương a_i b_i c_i d_i mô tả một đường một chiều nối nút giao thông (a_i,b_i) với nút (c_i,d_i) trên lưới. Mô tả các đường phố kết thúc bởi dòng gồm 4 số 0;
- Dòng cuối cùng ghi 4 số nguyên: u_s v_s u_t v_t là toạ độ của nút xuất phát (u_s, v_s) và nút kết thúc (u_t, v_t).

Kết quả: Ghi ra file văn bản có tên PATH.OUT đường đi ngắn nhất từ nút xuất phát đến nút kết thúc dưới dạng dãy các toạ độ của các nút cần phải đi qua bắt đầu từ nút xuất phát (u_s, v_s) và kết thúc ở nút (u_t, v_t) hoặc thông báo là không có đường đi nối hai nút.

Ví dụ: File PATH.INP và PATH.OUT có thể

PATH.INP
3 4 10 10 15 20 25 19 30 35 30 10 19 26 20 1 1 1 4 2 1 2 4 3 4 3 3 3 3 1 3 1 4 3 4 2 4 2 1 1 1 2 1 0 0 0 0 1 1 2 2

PATH.OUT
(1,1), (1,2), (1,3), (1,4), (2,4), (2,3), (2,2)

BA LÔ DU LỊCH

Tên chương trình: KNAPSACK.???

Ba lô du lịch mới của Steve được làm từ cao su siêu bền. Ba lô mới mang lại cho Steve những khả năng mới và cả những vấn đề mới.

Ba lô có dung lượng $V0$ cm^3 ($0 \le V0 \le 10^9$). Nếu đồ đạc mang theo có thể tích không quá $V0$ thì không có vấn đề gì xảy ra. Nhưng vì ba lô làm bằng cao su nên còn có thể nhét thêm nhiều thứ nữa, khi đó màng cao su sẽ căng và ép lên đồ vật bên trong. Nếu thể tích sử dụng là $V > V0$ thì các đồ vật trong ba lô sẽ phải chịu một áp lực $P = V - V0$.

Steve có n đồ vật có thể phải mang theo khi du lịch ($1 \le n \le 100$). Đồ vật thứ i có thể tích v_i, giá trị là c_i và chịu được áp lực không quá p_i.

Yêu cầu: Hãy xác định các đồ vật cần mang theo để tổng giá trị mang đi là lớn nhất.

Dữ liệu: Vào từ file văn bản KNAPSACK.INP:

- Dòng đầu tiên chứa 2 số nguyên n và $V0$,
- Dòng thứ i trong n dòng tiếp theo chứa 3 số nguyên c_i, v_i và p_i.

Kết quả: Đưa ra file văn bản KNAPSACK.OUT:

- Dòng đầu tiên chứa 2 số nguyên – số lượng đồ mang theo và tổng giá trị của chúng,
- Dòng thứ hai chứa các số nguyên chỉ ra một cách xác định các đồ cần mang theo.

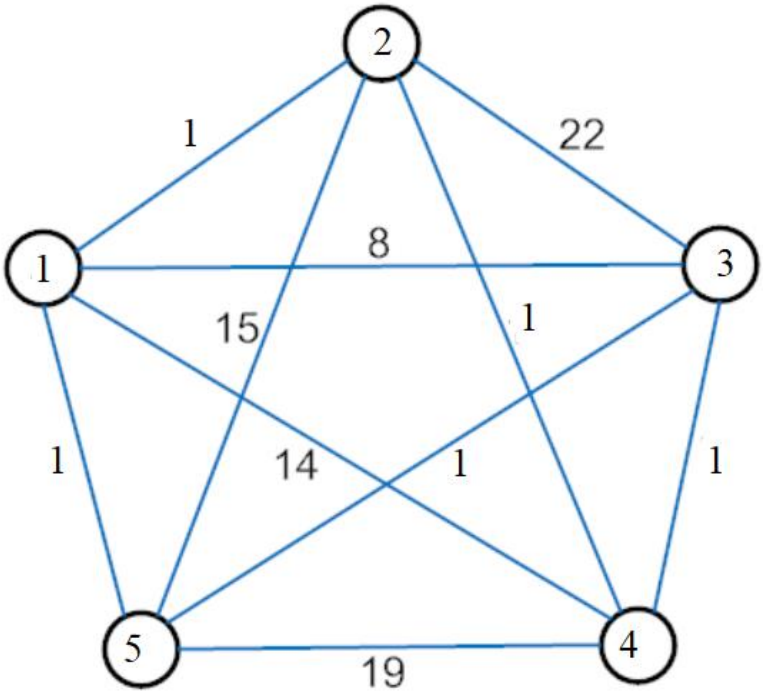
Ví dụ:

KNAPSACK.INP
3 10 3 1 1 4 1 2 5 1 3

KNAPSACK.OUT
2 2 2 3

BÀI TOÁN NGƯỜI THƯƠNG GIA

Bài toán kinh điển đã đưa ra hơn 200 năm trước: có N thành phố, khoảng cách giữa chúng được cho trước; người thương gia cần đi từ một thành phố, sau khi viếng thăm N-1 thành phố còn lại mỗi thành phố một lần và quay trở lại thành phố ban đầu. Trong hành trình của thương gia phải có độ dài nhỏ nhất.



Yêu cầu: Người thương gia khởi hành từ thành phố 1.Hãy cho biết hành trình có khoảng cách nhỏ nhất và in ra hành trình đó.

Dữ liệu: vào từ tập tin văn bản **TSP.INP**

- Dòng đầu tiên chứa số nguyên dương N ($1 \leq N \leq 20$) là số lượng thành phố .
- N dòng tiếp theo là mảng chiều A[NxN], giá trị A[i,j] ($0 \leq A[i,j] \leq 1000$) cho biết khoảng cách từ thành phố i đến thành phố j.

Kết quả: xuất ra tập tin văn bản **TSP.OUT**

- Dòng đầu tiên chứa một số nguyên là tổng nhỏ nhất hành trình tìm được.\
- Dòng thứ hai in ra hành trình của người thương gia

Ví dụ:

TSP.INP
5 **0 1 8 14 1** **1 0 22 1 15** **8 22 0 1 1** **14 1 1 0 19** **1 15 1 19 0**

TSP.OUT
5 **1 2 4 3 5 1**