

Mã đề thi: Đề Tin học_10

(Đề thi gồm 05 trang)

KỲ THI CHỌN HỌC SINH GIỎI

LẦN THỨ XVI, NĂM 2025

ĐỀ THI MÔN: Tin học - LỚP 10

Thời gian làm bài: 180 phút

(không kể thời gian giao đề)

Tổng quan đề thi

Tên bài	Tên file	Tên file INP	Tên file OUT	Thời gian	Điểm
Đưa thư	MAIL.*	MAIL.INP	MAIL.OUT	1s/test	7
Điệp viên	SPY.*	SPY.INP	SPY.OUT	1s/test	7
Truy tìm ngọc rồng	DRAGONBALL.*	DRAGONBALL.INP	DRAGONBALL.OUT	1s/test	6

Ghi chú: Dấu * tương ứng với đuôi .pas hoặc .cpp hoặc .py khi học sinh sử dụng ngôn ngữ lập trình PASCAL hoặc C++ hoặc Python

Đưa thư

Tại thị trấn Quahog, Rhode Island, các ngôi nhà đều được đánh dấu vị trí bằng một số nguyên trên trục tọa độ Ox và vị trí của các ngôi nhà là đôi một khác nhau. Có N ngôi nhà trong thị trấn này, ngôi nhà thứ i có tọa độ vị trí là x_i .

Người dân ở đây rất thích viết thư và muốn gửi chúng cho những người bạn của mình trong thành phố. Được biết, thời gian để bức thư gửi đi giữa hai nhà bằng đúng khoảng cách giữa hai nhà đó.

Với mỗi ngôi nhà thứ i trong thành phố, hãy tính Min_i và Max_i tương ứng với thời gian tối thiểu và thời gian tối đa nếu người ở ngôi nhà thứ i gửi thư cho bất kì ngôi nhà khác trong thị trấn.

Dữ liệu vào: từ tệp văn bản **MAIL.INP**:

- Dòng đầu tiên gồm số nguyên dương N ($2 \leq N \leq 10^5$), là số ngôi nhà trong thị trấn Quahog.
- Dòng thứ hai gồm N số nguyên $x_1, x_2, \dots, x_N$, $|x_i| \leq 10^9$, $i=1, \dots, n$, số nguyên x_i biểu diễn tọa độ vị trí của ngôi nhà thứ i trong thị trấn và vị trí của các ngôi nhà được sắp xếp theo thứ tự tăng dần.

Dữ liệu ra: đưa ra tệp văn bản **MAIL.OUT**:

- Gồm N dòng, dòng thứ i chứa 2 số nguyên là kết quả của Min_i và Max_i

Ví dụ:

MAIL.INP	MAIL.OUT
4	3 12
-5 -2 2 7	3 9
	4 7
	5 12

Ràng buộc:

- Subtask 1:** 40% số test có $N \leq 1000$;
- Subtask 2 :** 60% số test còn lại không có giới hạn gì thêm.

Điệp viên

Tại Albuquerque, một thành phố của New Mexico, Hank Schrader – đặc vụ của DEA, muốn tuyển cho mình 3 điệp viên để thâm nhập vào các tổ chức tội phạm.

Tại đây có N ứng viên tiềm năng, mỗi ứng viên có các mối quan hệ của mình, điều đó làm Hank rất khó để chọn lựa. Hank muốn chọn 3 điệp viên sao cho các điệp viên phải biết lẫn nhau từ trước và không nên biết quá nhiều người khác để tránh bị lộ thân phận. Với ứng viên, độ nhận diện của họ là số người họ biết (trừ 2 điệp viên còn lại).

Hãy giúp Hank chọn ra 3 điệp viên sao cho họ đều biết lẫn nhau và có tổng độ nhận diện của 3 người là nhỏ nhất.

Dữ liệu vào: từ tệp văn bản **SPY.INP**:

- Dòng đầu tiên gồm 2 số nguyên dương N và M ($3 \leq N \leq 5000$, $0 \leq M \leq 5000$), tương ứng là số ứng viên và số lượng mối quan hệ giữa các ứng viên;
- M dòng tiếp theo, dòng thứ i chứa 2 số nguyên a[i] và b[i] ($a[i] \neq b[i]$, $1 \leq a[i], b[i] \leq N$) với ý nghĩa ứng viên a[i] và b[i] đã biết nhau từ trước.

Dữ liệu ra: đưa ra tệp văn bản **SPY.OUT**:

- Gồm một dòng chứa số nguyên dương là tổng độ nhận diện của 3 điệp viên được chọn. Nếu không tìm được 3 điệp viên thỏa mãn, in ra -1.

Ví dụ:

SPY.INP	SPY.OUT
5 6	2
1 2	
1 3	
2 3	
2 4	
3 4	

4 5	
-----	--

Giải thích: 3 ứng viên được chọn là 1, 2, 3 với tổng độ nhận diện lần lượt là 0, 1, 1 (1 không quen ai ngoại trừ 2 điệp viên còn lại; 2 và 3 cùng quen 4)

Ràng buộc:

- **Subtask 1:** 50% số test có $N \leq 100$;
- **Subtask 2:** 50% số test còn lại không có giới hạn gì thêm.

Truy tìm ngọc rồng

Goku và Bulma đang trên đường tìm kiếm 7 viên ngọc rồng để cứu người bạn của mình. Goku đã có một số lượng ngọc rồng nhất định và phải tìm thêm K viên nữa. Khu vực Goku phải tìm kiếm gồm N thành phố và M con đường hai chiều nối liền các thành phố với nhau, con đường thứ i nối liền thành phố u[i] với thành phố v[i] và tốn thời gian di chuyển là w[i]. Trong đó các viên ngọc rồng nằm rải rác trong K thành phố. Nhờ vào radar của mình mà Bulma đã xác định được hết vị trí các viên ngọc rồng nằm ở thành phố nào.

Ban đầu Goku xuất phát từ thành phố 1, hãy giúp cậu ấy xác định thời gian ngắn nhất sao cho thu thập đủ K viên ngọc rồng để giải cứu bạn mình.

Lưu ý: Goku chỉ cần đến thành phố có ngọc rồng, thời gian lấy ngọc rồng không đáng kể.

Dữ liệu vào: từ tệp văn bản **DRAGONBALL.INP**:

- Dòng đầu tiên gồm 3 số nguyên dương N, K và M ($2 \leq N \leq 5000$; $1 \leq K \leq \min(7, N - 1)$; $M \leq 10000$) – Tương ứng là số thành phố, số lượng ngọc rồng phải tìm và số con đường trong thành phố.
- Dòng thứ 2 gồm K số nguyên p[i], số nguyên thứ i biểu diễn vị trí của viên ngọc rồng thứ i và không có 2 viên ngọc rồng nào nằm cùng vị trí ($2 \leq p[i] \leq N$).
- M dòng tiếp theo, mỗi dòng chứa 3 số nguyên u[i], v[i], w[i] tương ứng là có thể di chuyển từ u[i] đến v[i] trong thời gian w[i].

Dữ liệu ra: đưa ra tệp văn bản **DRAGONBALL.OUT**:

- Gồm một dòng chứa kết quả của bài toán.

Ví dụ:

DRAGONBALL.INP	DRAGONBALL.OUT
-----------------------	-----------------------

5 2 5	4
2 4	
1 2 2	
1 3 2	
2 3 1	
3 4 2	
1 4 1	

(Giải thích: di chuyển theo phương án sau sẽ tốn ít thời gian nhất: 1 -> 4 -> 1 -> 2)

Ràng buộc:

- **Subtask 1:** 20% số test có $K = 1$;
- **Subtask 2:** 20% số test có $N \leq 100$;
- **Subtask 3:** 60% số test còn lại không có giới hạn gì thêm.

.....Hết.....