

THỜI GIAN: 90 PHÚT
CHÚ Ý: SINH VIÊN ĐỌC KĨ ĐỀ NHẪM TÌM RA
CẤU TRÚC DỮ LIỆU PHÙ HỢP NHẤT

Bài 1 (4 điểm): Viết chương trình C nhận đầu vào là các thông tin được người dùng nhập như sau đây

Dòng 1, người dùng nhập số lượng các bạn trong nhóm. Chẳng hạn: 7

Trong các dòng tiếp theo, người dùng nhập các cặp bạn A, B mà đường đi trực tiếp từ nhà A đến nhà B (cũng như từ B đến A) bị cấm. Chẳng hạn nếu nhập 1 6 thì có nghĩa đường từ nhà bạn 1 đến nhà bạn 6 không thể đến trực tiếp

Nếu người dùng nhập \$ thì có nghĩa kết thúc việc nhập dữ liệu đầu vào. Ta có thể xem một ví dụ như sau:

7

1 6

2 3

5 4

0 6

\$

a) Yêu cầu người dùng nhập vào một con số X. Chương trình sẽ in ra tất cả các bạn mà X không đến trực tiếp nhà bạn ấy được. Trong trường hợp X đến trực tiếp được tất cả các nhà thì chương trình in ra -1. **(1 điểm)**

b) Yêu cầu người dùng nhập vào một con số Y. Chương trình sẽ in ra tất cả các đường đi xuất phát từ Y, đi qua tất cả nhà của các bạn MỘT LẦN. Trong trường hợp không tìm thấy giải pháp nào cả thì in ra -1. **(1 điểm)**

c) Chương trình sẽ phải in ra danh sách tất cả các đường đi giữa nhà các bạn sao cho thoả mãn yêu cầu có đường đi trực tiếp **(1 điểm)**. Chương trình cũng phải in ra tổng số các con đường này **(1 điểm)**. Chú ý rằng đường đi này không nhất thiết bắt đầu từ bạn số 0, mà có thể bắt đầu từ bạn nào cũng được (miễn là có trong danh sách)

Chẳng hạn, một trong số các đường đi xuất hiện trong danh sách (để cho) sẽ phải là:

0 4 3 1 5 6 7

Bài 2 (4 điểm): Viết chương trình C nhận đầu vào là các dòng liệt kê các tin nhắn giữa hai con người bất kỳ với nhau. Mỗi dòng có nội dung như sau:

DD-MM-YYYY sender size_of_message receiver blocked

Chẳng hạn:

03-12-2021 john 3.5M obama 0

Dòng trên có nghĩa là vào ngày 03 tháng 12 năm 2021, john đã gửi tin nhắn với dung lượng 3.5M đến với obama và tin nhắn này không bị hệ thống lọc thư rác chặn lại.

Khi người dùng nhấn dấu \$ thì chương trình sẽ ngừng việc nhập của người dùng lại.

a) Hãy kiểm tra xem trong số các dòng ở trên, liệu có bao nhiêu dòng thoả mãn yêu cầu về ngày tháng năm hợp lệ. Để đơn giản, ta giả sử không có ai gửi tin nhắn vào ngày 29/02 **(2 điểm)**

b) Hãy kiểm tra xem trong các tin nhắn đó, tin nhắn có dung lượng lớn nhất là bao nhiêu. Biết rằng nếu trong trường dung lượng tin nhắn (size_of_message) thì ký tự cuối cùng ám chỉ đến Gigabyte, Megabyte, Kilobyte lần lượt là G, M, K **(2 điểm)**

Bài 3 (2 điểm):

Tương tự với input như ở bài 2, cài đặt các chức năng sau:

a) Kiểm tra xem ngày nào là ngày có nhiều tin nhắn nhất giữa mọi người **(1 điểm)**

b) Yêu cầu người dùng nhập vào ngày tháng năm DD-MM-YYYY nào đó, cũng như tên người gửi A và người nhận B, chương trình kiểm tra xem trong ngày tháng năm đó, A đã gửi bao nhiêu tin nhắn cho B, tổng dung lượng A gửi B và bao nhiêu tin bị chặn lại bởi hệ thống **(1 điểm)**

Gợi ý:

Bài cuối sẽ cần phải cài đặt bảng băm để ra kết quả mau chóng khi mà bộ dữ liệu test có thể lên đến hàng triệu dòng.

Cụ thể hơn, ta cần một mảng, và mỗi phần tử của mảng là con trỏ trỏ đến nút gốc của cây nhị phân tìm kiếm.

Cách làm: xây dựng hàm nhận đầu vào là xâu ngày tháng int hash(DM), nó sẽ trả về vị trí của ngày tháng đó trong mảng X. Rõ ràng mảng X có 365 phần tử

Vậy khi đọc được một dòng dữ liệu thì chương trình sẽ gọi hàm băm để biết xem dòng dữ liệu đó sẽ được lưu vào phần tử chỉ số bao nhiêu của mảng.

Với mỗi dòng dữ liệu, ta tạo ra một nút với key của nút đó là xâu tạo thành từ sender, receiver và năm.

31-05-2022 kien 2.3M viet 0

thì ta tạo ra một nút trong cây nhị phân với từ khoá key = “kien_viet_2022”.

nút đó còn có thông tin về dung lượng đã gửi và số tin đã bị chặn.

Kiểm tra xem cây nhị phân tìm kiếm đã có nút với key “kien_viet_2022” chưa? Nếu rồi thì ta tăng giá trị của trường dữ liệu lên. Cụ thể, ta tăng dung lượng đã gửi lên thêm 2.3M; số tin đã bị chặn không tăng lên.