

HƯỚNG DẪN THUẬT TOÁN ĐỀ THI HỌC SINH GIỎI TIN HỌC CẤP TỈNH BẢNG B – QUẢNG NINH NĂM 2022

SỞ GIÁO DỤC VÀ ĐÀO TẠO KỲ THI CHỌN HỌC SINH GIỎI CẤP TỈNH THPT NĂM 2022
TỈNH QUẢNG NINH

Môn thi: TIN HỌC - Bảng B

ĐỀ THI CHÍNH THỨC

Ngày thi: 02/12/2022

Thời gian làm bài: 180 phút, không kể thời gian giao đề

(Đề thi này có 04 trang)

TỔNG QUAN ĐỀ THI

Bài	Tên bài	Tệp chương trình	Tệp dữ liệu	Tệp kết quả	Bộ nhớ	Thời gian / test	Điểm
1	Số đặc biệt	prime.*	prime.inp	prime.out	1024 MB	1 giây	6
2	Luyện thi	practice.*	practice.inp	practice.out	1024 MB	1 giây	6
3	Mật khẩu	password.*	password.inp	password.out	1024 MB	1 giây	5
4	Lát nền	tili.*	tili.inp	tili.out	1024 MB	1 giây	3

Dấu * được thay thế bởi pas hoặc cpp hoặc py của ngôn ngữ lập trình được sử dụng tương ứng là Pascal hoặc C++ hoặc Python.

Hãy lập trình giải các bài toán sau:

Bài 1. Số đặc biệt (6 điểm)

Minh là một người rất yêu thích số nguyên tố, đồng thời cậu cũng rất thích số 5. Do đó, cậu ta luôn coi những số nguyên tố có tổng các chữ số chia hết cho 5 là số đặc biệt. Lần này, thầy giáo cho Minh hai số nguyên dương L, R ($L \leq R$) và muốn Minh cho biết trong đoạn $[L, R]$ có bao nhiêu số đặc biệt. Bạn hãy lập trình tìm câu trả lời giúp Minh.

Dữ liệu vào: Vào từ tệp văn bản prime.inp có cấu trúc như sau:

- Dòng đầu tiên chứa số nguyên dương $T \leq 10^5$ là số lượng cặp số nguyên L, R cần trả lời;
- T dòng tiếp theo, mỗi dòng chứa hai số nguyên L, R ($0 < L \leq R \leq 3 \times 10^6$) phân biệt nhau bởi một dấu cách.

Dữ liệu ra: Ghi ra tệp văn bản prime.out gồm T dòng, mỗi dòng ghi một số nguyên là số lượng số đặc biệt trong đoạn $[L, R]$ tương ứng với thứ tự dữ liệu đầu vào.

Ví dụ:

prime.inp	prime.out
2	1
1 10	2
4 20	

Giải thích: Đoạn $[1, 10]$ có 1 số đặc biệt là 5, đoạn $[4, 20]$ có 2 số đặc biệt là 5 và 19.

Ràng buộc:

- Có 40% số test ứng với 40% số điểm của bài thỏa mãn: $T = 1; 0 < L, R \leq 10^3$;
- 30% số test khác ứng với 30% số điểm của bài thỏa mãn: $2 \leq T \leq 10; 0 < L, R \leq 10^5$;
- 30% số test còn lại ứng với 30% số điểm của bài thỏa mãn: $T \leq 10^5; 0 < L, R \leq 3 \times 10^6$;

Bài 2. Luyện thi (6 điểm)

Để chuẩn bị cho hội thi Tin học trẻ sắp tới, thầy giáo quyết định giao một bộ đề thi gồm N đề thi khác nhau cho Minh để tự luyện tập. Các đề thi được đánh số từ 1 đến N , mỗi đề thi sẽ giúp rèn luyện một số kỹ năng cho Minh để có thể đạt thành tích tốt trong hội thi Tin học trẻ.

Nhằm định hướng cho quá trình luyện tập đạt hiệu quả cao nhất, mỗi đề thi có một yêu cầu tối thiểu về trình độ kỹ năng. Đề giải được đề thi i , Minh cần có trình độ kỹ năng tối thiểu là a_i . Điều này có nghĩa là Minh có thể giải được đề thi thứ i khi và chỉ khi có trình độ kỹ năng bằng hoặc lớn hơn a_i . Nếu giải được đề thi i thì trình độ kỹ năng của Minh sẽ tăng thêm một lượng là b_i . Giả sử ban đầu, trình độ kỹ năng của Minh trước khi luyện tập là c . Các đề thi có thể được giải theo trình tự bất kỳ.

Yêu cầu: Cho các số nguyên N, c và các cặp giá trị (a_i, b_i) , hãy xác định số lượng đề thi tối đa mà Minh có thể giải được.

Dữ liệu vào: Vào từ tệp văn bản `practice.inp` có cấu trúc như sau:

- Dòng đầu tiên chứa hai số nguyên N, c ($1 \leq N \leq 10^5, 0 \leq c \leq 10^9$);
- Dòng thứ i trong N dòng tiếp theo ($1 \leq i \leq N$) chứa 2 số nguyên a_i và b_i ($1 \leq a_i, b_i \leq 10^9$). Các số trên cùng một dòng được phân cách nhau bởi một dấu cách.

Dữ liệu ra: Ghi ra tệp văn bản `practice.out` một số nguyên dương, là số đề thi tối đa mà Minh có thể giải được.

Ví dụ:

<code>practice.inp</code>	<code>practice.out</code>
4 1	3
1 10	
21 5	
1 10	
100 100	

Giải thích: Với $N = 4, c = 1$ và các cặp giá trị (a_i, b_i) tương ứng là $(1, 10), (21, 5), (1, 10), (100, 100)$, Minh sẽ giải đề thi 1 sau đó giải đề thi 3 và cuối cùng giải đề thi 2. Đề thi 4 Minh không giải được vì không đủ kỹ năng. Như vậy, Minh giải được tối đa 3 đề thi.

Ràng buộc:

- Có 60% số test ứng với 60% số điểm của bài thỏa mãn: $N \leq 1000, (1 \leq a_i, b_i \leq 10^5)$;
- 40% số test còn lại ứng với 40% số điểm của bài không có thêm ràng buộc nào.

Bài 3. Mật khẩu (5 điểm)

Do ảnh hưởng của đại dịch Covid-19, ban tổ chức hội thi Tin học trẻ dự kiến sẽ cho các thí sinh thi theo hình thức trực tuyến. Để tham gia thi, thí sinh sẽ được cung cấp một tài khoản và mật khẩu để đăng nhập. Để đảm bảo yếu tố bảo mật, ban tổ chức yêu cầu các thí sinh phải thay đổi mật khẩu ngay sau khi nhận được. Mật khẩu được coi là “an toàn” nếu thỏa mãn đồng thời các điều kiện sau:

- Có độ dài tối thiểu là 6;
- Chứa ít nhất 1 chữ cái in hoa ('A', 'B', ..., 'Z');
- Chứa ít nhất 1 chữ cái in thường ('a', 'b', ..., 'z');
- Chứa ít nhất 1 chữ số ('0', '1', ..., '9').

Ví dụ các xâu kí tự: “a1B2C3”, “Tinhoc12” là các mật khẩu an toàn, còn các xâu kí tự: “a1B2C”, “a1b2c3”, “A1B2C3”, “TinHoc” không phải là mật khẩu an toàn.

Cho một xâu kí tự S chỉ gồm các loại kí tự: chữ cái in hoa, chữ cái in thường và chữ số. Hãy cho biết có bao nhiêu cặp chỉ số (i, j) thỏa mãn đồng thời hai điều kiện:

- $1 \leq i < j \leq |S|$, trong đó $|S|$ là độ dài xâu S , $0 \leq |S| \leq 10^5$;
- Xâu con S_{ij} gồm các kí tự liên tiếp từ chỉ số i đến j của xâu S là mật khẩu an toàn.

Yêu cầu: Cho xâu kí tự S , tính số lượng cặp (i, j) thỏa mãn hai điều kiện trên.

Dữ liệu vào: Vào từ tệp văn bản `password.inp` gồm một dòng chứa xâu kí tự S .

Dữ liệu ra: Ghi ra tệp văn bản `password.out` một số nguyên dương là số lượng cặp chi số (i, j) thỏa mãn điều kiện xâu con S_{ij} là mật khẩu an toàn.

Ví dụ:

<code>password.inp</code>	<code>password.out</code>
abc123	0
abc3456789PO	6

Ràng buộc:

- Có 40% số test ứng với 40% số điểm của bài thỏa mãn: $0 \leq |S| \leq 10^3$;
- 60% số test còn lại ứng với 60% số điểm của bài không có thêm ràng buộc nào.

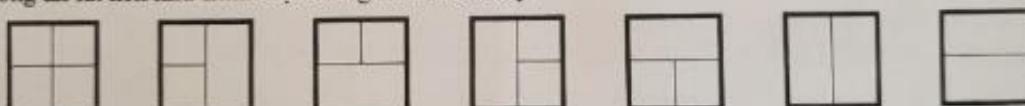
Bài 4. Lát nền (3 điểm)

Viện Công nghệ thông tin đang được tu sửa và nâng cấp. Một trong những hạng mục công việc là lát lại hành lang nối từ phòng làm việc sang phòng đặt máy chủ. Hành lang có độ rộng 2 và độ dài n được biểu thị như một lưới ô vuông gồm 2 hàng và n cột. Để lát người ta dùng các viên gạch men loại kích thước 1×1 và kích thước 1×2 với số lượng dự trữ không hạn chế. Các viên gạch 1×2 có thể lát dọc hoặc xoay ngang. Trước đây hành lang được lát bằng các viên gạch kích thước 1×1 và có k ($k = 0$ hoặc $k = 1$) viên gạch bên dưới lắp các thiết bị điện tử. Nếu $k = 1$ thì viên gạch đó ở hàng r và cột c . Ban Giám đốc viện không muốn lặp lại hệ thống điện tử vốn đang hoạt động rất tốt, nên yêu cầu đánh dấu viên gạch này và không được bóc lên trong quá trình lát nền.

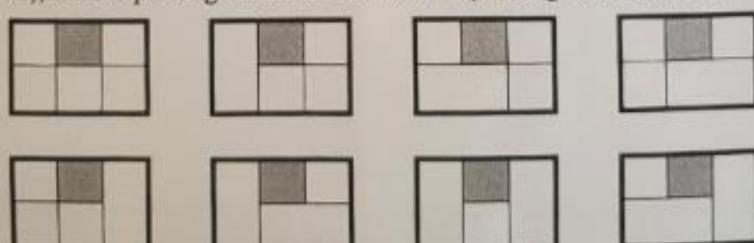
Bộ phận thi công phản nản về yêu cầu trên, vì như thế sẽ hạn chế khả năng lát. Điều này làm Trường phòng vật tư đề nghị bộ phận lập trình tính số phương án lát nền khác nhau mà vẫn đảm bảo yêu cầu đã nêu, để bên thi công thấy có nhiều cách làm khác nhau.

Bạn hãy tính và đưa ra số phương án lát nền theo mô-đun $10^9 + 7$ (tức là đưa ra số dư của số phương án lát nền chia cho $10^9 + 7$). Hai phương án gọi là khác nhau nếu tồn tại hai ô kề cạnh trong phương án này được phủ bằng một viên gạch 1×2 , còn theo phương án kia thì không được phủ bằng một viên gạch 1×2 .

Ví dụ với $n = 2$ và $k = 0$ (không có viên gạch kích thước 1×1 nào được đánh dấu), ta có 7 phương án lát nền như minh họa trong hình dưới đây:



Ví dụ khác với $n = 3$ và $k = 1$ viên gạch kích thước 1×1 được đánh dấu ở vị trí $(1, 2)$ (ô được tô kín trong hình vẽ), ta có 8 phương án lát nền như minh họa trong hình dưới đây:



Dữ liệu: Vào từ tệp văn bản `tili.inp`. Dòng đầu tiên chứa số nguyên n và k ($1 \leq n \leq 10^5$; $k = 0$ hoặc $k = 1$). Nếu $k = 1$ thì dòng tiếp theo chứa hai số nguyên r và c ($1 \leq r \leq 2$; $1 \leq c \leq n$).

Kết quả: Ghi ra tệp văn bản `tili.out` một số nguyên là số phương án lát nền theo mô-đun $10^9 + 7$.

Ví dụ:

<code>tili.inp</code>	<code>tili.out</code>
2 0	7
3 1	8
1 2	

Ràng buộc:

- Có 25% số test ứng với 25% số điểm của bài thỏa mãn: $1 \leq n \leq 8; k = 0$;
- 25% số test khác ứng với 25% số điểm của bài thỏa mãn: $1 \leq n \leq 10^3; k = 0$;
- 25% số test khác ứng với 25% số điểm của bài thỏa mãn: $1 \leq n \leq 10^5; k = 0$;
- 25% số test còn lại ứng với 25% số điểm của bài không có thêm ràng buộc nào.

----- HẾT -----

Họ và tên thí sinh: Hoàng Trung Mẫn Số báo danh: 2055

Chữ ký của Giám thị 1: MT Chữ ký của Giám thị 2: TL

Hướng dẫn thuật toán

Bài 1. Số đặc biệt

- a) Sub 1: Vì $T = 1, 0 < L < R < 10^3$. Nên bài toán trở thành tìm các số trong đoạn $[L, R]$ thỏa mãn yêu cầu bài toán.

Ta có thuật toán đơn giản như sau:

```
int cnt = 0; //Biến cnt để lưu số các số đặc biệt (thỏa mãn yêu cầu bài toán)
```

```
For(int i=L; i<=R; i++){ //Duyệt các số i từ L đến R
```

- Với mỗi số i ta kiểm tra số i có phải là số nguyên tố hay không

//Chúng ta có giải thuật với độ phức tạp $O(\sqrt{i})$ để kiểm tra i có phải số nguyên tố hay không

- Nếu i là số nguyên tố ta tính tiếp tổng S các chữ số của i

Nếu S chia hết cho 5 thì tăng biến đếm lên một đơn vị, cnt = cnt+1;

}

Thuật toán có độ phức tạp $O([R - L] * \sqrt{R})$. Trường hợp xấu nhất $R = 10^3$ thuật toán vẫn thỏa mãn.

- b) Sub 2: Thuật toán 2: Có $2 \leq T \leq 10$; $0 < L < R < 10^5$.

Nếu ta áp dụng thuật toán 1 cho mỗi truy vấn thì ta có thuật toán như sau:

```
For(int t= 1; t<=T; t++){
```

Int L, R;

Cin>>L>>R;

```
int cnt = 0; //Biến cnt để lưu số các số đặc biệt (thỏa mãn yêu cầu bài toán)
```

```
For(int i=L; i<=R; i++){ //Duyệt các số i từ L đến R
```

- Với mỗi số i ta kiểm tra số i có phải là số nguyên tố hay không

//Chúng ta có giải thuật với độ phức tạp $O(\sqrt{i})$ để kiểm tra i có phải số nguyên tố hay không
- Nếu i là số nguyên tố ta tính tiếp tổng S các chữ số của i

Nếu S chia hết cho 5 thì tăng biến đếm lên một đơn vị, cnt = cnt+1;

}

}

Độ phức tạp là $O(T * [R - L] * \sqrt{R})$

$T * [R - L] * \sqrt{R} \leq 10 * 10^5 * \sqrt{10^5} \leq 3 * 10^8$ nên thuật toán sẽ thất bại với dữ kiện của bài toán

Vì vậy ở đây chúng ta sẽ cải tiến thuật toán bằng cách sử dụng sàng nguyên tố. Gọi snt[i] bằng true nếu i là số nguyên tố và snt[i] bằng false nếu i không phải là số nguyên tố. Khi đó, chúng ta có thuật toán cải tiến như sau:

```
For(int t= 1; t<=T; t++){  
    Int L, R;  
    Cin>>L>>R;  
    int cnt = 0; //Biên cnt để lưu số các số đặc biệt (thỏa mãn yêu cầu bài toán)  
    For(int i=L; i<=R; i++){ //Duyệt các số i từ L đến R  
        //S là tổng các chữ số của i  
        If(snt[i]==true and S%5==0) cnt++;  
    }  
}
```

Thuật toán có độ phức tạp: $O(T * (R - L) + 10^6 * \log_2 10^6)$. Trong đó, $10^6 * \log_2 10^6$ là chi phí để làm sàng nguyên tố.

c) Sub 3: $2 \leq T \leq 10^5$; $0 < L < R < 10^5$. Thuật toán 3:

Nếu ta áp dụng thuật toán 2 thì độ phức tạp $T * (R - L) \leq 10^5 * 10^5 = 10^{10}$ nên thuật toán sẽ bị quá 1s.

Vì vậy chúng ta cần cải tiến thuật toán bằng kĩ thuật sử dụng tổng cộng dồn như sau:

Bước 1: Chuẩn bị trước sàng nguyên tố, snt[i]==true nếu i là số nguyên tố, snt[i]==false nếu i không phải là số nguyên tố.

Chuẩn bị trước mảng S[], S[i] là tổng các chữ số của i

Gọi F[i] là số các số đặc biệt trong phạm vi từ 1 đến i. Ta tính mảng F[i] như sau:

```
For(int i=1; i<=10^6; i++){  
    F[i] = F[i-1];  
    If(snt[i]==true and S[i]%5==0) F[i]++;  
}
```

Bước 2: Dùng tổng cộng dồn F[i] để cải tiến thuật toán số 2 như sau:

```
For(int t= 1; t<=T; t++){
```

```
Int L, R;  
Cin>>L>>R;  
int cnt = 0; //Biến cnt để lưu số các số đặc biệt (thỏa mãn yêu cầu bài toán)  
cnt = F[R] - F[L-1];  
cout<<cnt<<endl;  
}
```