

ÔN TẬP – THUẬT TOÁN TÌM KIẾM TUẦN TỰ

Bài tập 0: Bạn đã được cung cấp một mảng có kích thước N bao gồm các số nguyên. Ngoài ra, bạn đã được cung cấp một phần tử M, bạn cần tìm và in chỉ mục của lần xuất hiện cuối cùng của phần tử M này trong mảng nếu nó tồn tại trong đó, nếu không thì in -1. Xem xét mảng này là 1 chỉ mục.

Dữ liệu vào: Ghi trong tệp văn bản

Dòng đầu tiên bao gồm 2 số nguyên N và M biểu thị kích thước của mảng và phần tử được tìm kiếm trong mảng tương ứng. Dòng tiếp theo chứa N số nguyên được phân tách bằng dấu cách biểu thị các phần tử của mảng.

Dữ liệu ra: Ghi ra tệp văn bản

In một số nguyên duy nhất biểu thị chỉ số của lần xuất hiện cuối cùng của số nguyên M trong mảng nếu nó tồn tại, nếu không thì in -1.

Ví dụ:

LSEARCH.INP	LSEARCH.INP
5 1 1 2 3 4 1	5

Ý tưởng:

Đối với lần xuất hiện cuối cùng của phần tử khóa, hãy tìm kiếm từ phải sang trái thay vì di chuyển từ trái sang phải.

Bài tập 1: Tính tổng của một tập hợp

Chúng ta định nghĩa 'tập hợp' là tập hợp các số riêng biệt. Đối với hai tập hợp A và B, chúng ta xác định tập hợp tổng của chúng là tập $S(A, B) = \{a + b \mid a \in A, b \in B\}$. Nói cách khác, tập $S(A, B)$ chứa tất cả các phần tử có thể được biểu diễn dưới dạng tổng của một phần tử trong A và một phần tử trong B. Cho hai tập A, C, nhiệm vụ của bạn là tìm tập B có số nguyên dương nhỏ hơn hoặc bằng 100 với kích thước tối đa sao cho $S(A, B) = C$. Nó được đảm bảo rằng có duy nhất bộ như vậy.

Dữ liệu vào: Cho trong tệp văn bản **SUMSET.INP** gồm

- Dòng đầu tiên chứa N biểu thị số phần tử trong tập A, dòng sau chứa N số nguyên được phân tách bằng dấu cách ai biểu thị các phần tử của tập A.
- Dòng thứ ba chứa M biểu thị số phần tử trong tập C, dòng sau chứa M số nguyên được phân tách bằng dấu cách ci biểu thị các phần tử của tập C.

Dữ liệu ra: Ghi ra tệp văn bản **SUMSET.OUT** - In tất cả các phần tử của B theo thứ tự tăng dần trong một dòng, cách nhau bởi khoảng trắng.

Ví dụ:

SUMSET.INP	SUMSET.OUT
2 1 2 3 3 4 5	2 3

Ràng buộc:

- $1 \leq N, M \leq 100$
- $1 \leq a_i, c_i \leq 100$

Nếu e là một phần tử của tập B, thì $e + 2$ là một phần tử của tập C, vì vậy chúng ta phải có $e \leq 3$. Rõ ràng, e không thể là 1 vì $1 + 1 = 2$ không phải là phần tử của tập C. Do đó, $B = \{2, 3\}$.

Giải thuật:

Như trong câu lệnh, các trường hợp kiểm thử được tạo để đảm bảo tồn tại một tập hợp duy nhất B. Vì vậy, với mỗi số từ 1 đến 100, chúng tôi sẽ cố gắng thêm nó vào tập B nếu có thể. Hãy xem trường hợp chúng ta không thể thêm số e để đặt B:

Nếu e ở trong tập B, thì với mỗi phần tử ai trong tập A, chúng ta phải có $e + a_i$ là một phần tử của tập C. Vì vậy, chúng ta không thể thêm e vào tập B nếu tồn tại một phần tử ai sao cho $e + a_i$ Không phải là một phần tử của tập C. Đó là tất cả những gì chúng ta cần.

Mã giả bên dưới:

for e from 1 to 100:

```
{
    can_add = true
    for a in A:
    {
        if e + a is not an element of C:
        {
            can_add = false
        }
    }
    if can_add:
    {
        print(e)
    }
}
```

Bài tập 2: Đồ áo

Chúng ta có m bộ quần áo ướt hoàn toàn được phơi dưới ánh nắng mặt trời chờ đợi để khô. Bây giờ chúng ta đang ở giây thứ t_1 và trời đang mưa. Trời sẽ mưa trở lại vào giây $t_2 \dots t_n$ và sau mỗi cơn mưa quần áo sẽ lại ướt hoàn toàn. Bộ đồ thứ i cần a_i giây để phơi quần áo để được khô. Chúng ta có thể ra ngoài và thu thập tất cả quần áo khô bất cứ lúc nào nhưng không thể làm điều này nhiều hơn g lần. Số lượng quần áo tối đa chúng ta có thể thu thập cho đến lần thứ hai là bao nhiêu? Lưu ý rằng thời gian của mỗi cơn mưa gần như bằng không, vì vậy chúng ta có thể bỏ qua nó. Ngoài ra việc thu thập quần áo khô không mất bất kỳ thời gian từ nào.

Dữ liệu vào: Cho trong tệp văn bản **CLOTHES.INP** gồm

- Dòng đầu tiên chứa ba số nguyên n, m, g ($2 \leq n \leq 100$, $1 \leq m, g \leq 100$) tương ứng.
- Dòng thứ hai sẽ là n số tăng biểu thị $t_1 \dots t_n$ ($0 \leq t_1 \dots t_n \leq 10^4$).
- Dòng cuối cùng sẽ có số m biểu thị $a_1 \dots a_m$ ($1 \leq a_i \leq 10000$).

Dữ liệu ra: Ghi ra tệp văn bản **CLOTHES.OUT** gồm một dòng in số lượng tối đa của quần áo chúng ta có thể thu thập.

Ví dụ:

CLOTHES.INP	CLOTHES.OUT
3 3 2 3 5 8 4 1 3	2

Giải thích:

Trong ví dụ mẫu, đầu tiên chúng tôi đi ra ngoài vào thời điểm 5 (chính xác là trước khi mưa) và lấy mảnh quần áo thứ hai. Sau đó, lần thứ hai, vào lần thứ 8, chúng tôi đi và lấy mảnh thứ ba.

Bài tập 3: Nhà sư đi bộ

Hôm nay, Nhà sư đi dạo trong một khu vườn. Có rất nhiều cây trong vườn và mỗi cây có một bảng chữ cái tiếng Anh trên đó. Trong khi nhà sư đang đi bộ, anh nhận thấy rằng tất cả các cây có nguyên âm trên đó đều không ở trạng thái tốt. Anh quyết định chăm sóc chúng. Vì vậy, anh ấy yêu cầu bạn nói cho anh ấy đếm những cây như vậy trong vườn.

Lưu ý: Các chữ cái sau đây là nguyên âm: 'A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o' và 'u'.

Dữ liệu vào: Cho trong tệp văn bản **MWALK.INP** gồm

Dòng đầu tiên bao gồm một số nguyên T biểu thị số lượng các trường hợp thử nghiệm. Mỗi trường hợp kiểm tra chỉ bao gồm một chuỗi, mỗi ký tự của chuỗi biểu thị bằng chữ cái (có thể là chữ thường hoặc chữ hoa) trên một cây trong vườn.

Dữ liệu ra: Ghi ra tệp văn bản **MWALK.INP** gồm - Đối với mỗi trường hợp thử nghiệm, in số đếm trong một dòng mới.

Ví dụ:

MWALK.INP	MWALK.OUT
2	2
nBBZLaosnm	1
JHkIsnZtTL	

Giải thích: Trong trường hợp thử nghiệm 1, a và o là nguyên âm duy nhất. Vậy, đếm = 2

Ràng buộc (Constraints):

- $1 \leq T \leq 10$
- $1 \leq \text{length of string} \leq 10^5$

Mô tả tóm tắt: Cho một chuỗi S, bạn phải đếm số nguyên âm trong chuỗi.

Giải thuật chi tiết:

Phương pháp 1: Đối với mỗi nguyên âm, hãy đếm số lần xuất hiện trong chuỗi S. Câu trả lời cuối cùng sẽ là tổng tần số của tất cả các nguyên âm.

Phương pháp 2: Lặp lại tổng thể tất cả các ký tự trong chuỗi S và sử dụng bộ đếm (biến) để theo dõi số nguyên âm trong chuỗi S. Trong khi lặp lại các ký tự, nếu chúng ta gặp nguyên âm, chúng ta sẽ tăng bộ đếm lên 1.

Độ phức tạp thời gian: $O(N)$ trong đó N là độ dài của chuỗi S.

Độ phức tạp không gian: $O(N)$

Bài tập 4: Max_Min

Cho một dãy N số nguyên dương $a_1, a_2, a_3, \dots, a_n$, Tìm các giá trị nhỏ nhất và giá trị lớn nhất có thể được tính bằng cách tính tổng chính xác N-1 của N số nguyên. Sau đó in các giá trị nhỏ nhất và giá trị lớn nhất tương ứng dưới dạng trên một dòng gồm hai số nguyên dài phân cách nhau một khoảng trắng.

Dữ liệu vào: Cho trong tệp văn bản **MAXMIN.INP** gồm

- Dòng đầu tiên lấy giá trị đầu vào của N
- Dòng thứ hai Lấy giá trị số nguyên phân tách N không gian

Dữ liệu ra: Ghi ra tệp văn bản **MAXMIN.OUT** gồm hai giá trị tách biệt (một tổng tối đa và một tổng tối thiểu)

Ví dụ:

MAXMIN.INP	MAXMIN.OUT
5	10 14
1 2 3 4 5	

Ràng buộc: $0 < N < 100001$; $0 \leq a_i < 10^{13}$

Giải thích: Các số ban đầu của chúng tôi là 1,2,3,4 và 5. Chúng tôi có thể tính các khoản tiền sau bằng bốn trong số năm số nguyên:

- Nếu chúng ta tổng hợp tất cả mọi thứ trừ 1, tổng của chúng tôi là $2 + 3 + 4 + 5 = 14$.
- Nếu chúng ta tổng hợp tất cả mọi thứ trừ 2, tổng của chúng tôi là $1 + 3 + 4 + 5 = 13$.
- Nếu chúng ta tổng hợp tất cả mọi thứ trừ 3, tổng của chúng tôi là $1 + 2 + 4 + 5 = 12$.
- Nếu chúng ta tổng hợp tất cả mọi thứ trừ 4, tổng của chúng tôi là $1 + 3 + 4 + 5 = 11$.
- Nếu chúng ta tổng hợp tất cả mọi thứ trừ 5, tổng của chúng tôi là $1 + 2 + 3 + 4 = 10$.

Như bạn có thể thấy, tổng tối thiểu là $1 + 2 + 3 + 4 = 10$ và tổng tối đa là $2 + 3 + 4 + 5 = 14$. Do đó, chúng tôi in các khoản tiền tối thiểu và tối đa này dưới dạng hai số nguyên được phân tách bằng dấu cách trên một dòng mới.

Giải thuật:

Chỉ cần tìm tổng của tất cả các yếu tố và hiển thị (Tổng tất cả các yếu tố - Yếu tố tích cực nhất) và (Tổng hợp tất cả các yếu tố - Yếu tố tiêu cực nhất)

Bài tập 5: Hình chữ nhật vàng

Bạn có N hình chữ nhật. Một hình chữ nhật là vàng nếu tỷ lệ các cạnh của nó nằm trong khoảng $[1.6, 1.7]$, bao gồm cả hai. Nhiệm vụ của bạn là tìm số lượng hình chữ nhật vàng.

Dữ liệu vào: Cho trong tệp văn bản **RGOLDEN.INP** gồm

- Dòng đầu tiên: Số nguyên N biểu thị số lượng hình chữ nhật
- Mỗi dòng N sau: Hai số nguyên W, H biểu thị chiều rộng và chiều cao của hình chữ nhật.

Dữ liệu ra: Ghi ra tệp văn bản **RGOLDEN.OUT** - In câu trả lời trong một dòng duy nhất.

Ví dụ:

RGOLDEN.INP	RGOLDEN.OUT
5 10 1 165 100 180 100 170 100 160 100	3

Ràng buộc (Constraints): $1 \leq N \leq 10^5$; $1 \leq W, H \leq 10^9$

Giải thích: Có ba hình chữ nhật vàng: (165, 100), (170, 100), (160, 100).

Giải thuật: Vấn đề này rất đơn giản, chỉ cần kiểm tra xem w/h hoặc h/w có nằm trong khoảng $[1.6, 1.7]$ không. Một điều cần lưu ý ở đây là chúng ta nên sử dụng loại số nguyên thay vì loại kiểu số thực. Nó an toàn hơn. Vì vậy, điều kiện trở thành: $16 \times w \leq 10 \times h \leq 17 \times w$ hoặc $16 \times h \leq 10 \times w \leq 17 \times h$.

Độ phức tạp thời gian: $O(N)$.

Bài tập 6: Đếm con đường ếch đi

Có một con ếch ban đầu được đặt ở gốc của mặt phẳng tọa độ. Trong đúng 1 giây, con ếch có thể di chuyển lên 1 đơn vị, di chuyển sang phải 1 đơn vị hoặc đứng yên. Nói cách khác, từ vị trí (x, y), ếch có thể dành 1 giây để di chuyển đến:

- (x + 1, y)
- (x, y + 1)
- (x, y)

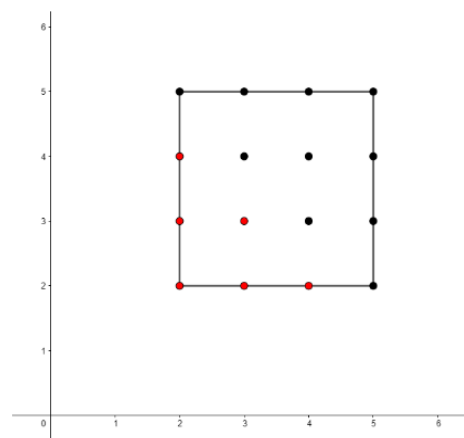
Sau T giây, một dân làng nhìn thấy con ếch báo cáo rằng con ếch nằm trên hoặc bên trong một hình vuông có độ dài cạnh có tọa độ (X, Y), (X + s, Y), (X, Y + s), (X + s, Y + s). Tính xem có bao nhiêu điểm có tọa độ nguyên trên hoặc bên trong hình vuông này có thể là vị trí của ếch sau chính xác T giây.

Dữ liệu vào: Cho trong tệp văn bản **FROG.INP** gồm

Dòng đầu tiên và duy nhất chứa bốn số nguyên được phân tách bằng dấu cách: X, Y, s và T.

Dữ liệu ra: Ghi ra tệp văn bản **FROG.OUT** - In số lượng điểm có tọa độ nguyên có thể là vị trí của ếch sau T giây.

Ví dụ:



FROG.INP	FROG.OUT
2 2 3 6	6

Ràng buộc:

- $0 \leq X, Y \leq 100$;
- $1 \leq s \leq 100$;
- $0 \leq T \leq 400$

Giải thích: Các điểm được hiển thị là các điểm trên hoặc trong ô vuông đã chỉ định và các điểm có màu đỏ là các điểm mà ếch có thể ở sau 6 giây.

Giải thuật: Vì giới hạn cho s là thấp, nên một cách tiếp cận vũ phu sẽ hoạt động. Chúng ta có thể lặp qua mọi tọa độ trên hoặc bên trong hình vuông đã cho, rồi kiểm tra xem con ếch có thể đạt đến điểm này hay không.

Bây giờ chúng ta cần trả lời câu hỏi, phải mất bao nhiêu giây để con ếch đạt đến điểm (x, y)? Con ếch sẽ thực hiện x bước sang phải và y bước lên, vì vậy đó là x + y bước cùng nhau. Do đó, để kiểm tra xem con ếch có thể đạt đến điểm (x, y) hay không, chúng ta chỉ cần kiểm tra xem $x + y \leq T$. Vì vậy, chúng ta có thể giữ một bộ đếm và lặp qua tất cả các điểm để giải quyết vấn đề.

Bài tập 7: Tổng lớn nhất

Bạn được cung cấp một mảng các số nguyên A, bạn cần tìm tổng tối đa có thể thu được bằng cách chọn một số tập hợp con không trống của mảng. Nếu có nhiều tập con không trống như vậy, hãy chọn tập con có số phần tử tối đa. In tổng tối đa và số phần tử trong tập hợp con đã chọn.

Dữ liệu vào: Cho trong tệp văn bản **SUBSET.INP** gồm

Dòng đầu tiên chứa một số nguyên N, biểu thị số phần tử của mảng. Dòng tiếp theo chứa N số nguyên được phân tách bằng dấu cách, biểu thị các phần tử của mảng.

Dữ liệu ra: Ghi ra tệp văn bản **SUBSET.OUT** gồm

- In 2 số nguyên được phân tách bằng dấu cách, tổng tối đa có thể thu được bằng cách chọn một số tập hợp con và số phần tử tối đa trong số tất cả các tập hợp con có cùng tổng tối đa.

Ví dụ:

SUBSET.INP	SUBSET.OUT
5 1 2 -4 -2 3	6 3

Giải thích: Tập hợp con được chọn là {1, 2, 3}.

Ràng buộc:

- $1 \leq N \leq 10^5$;
- $-109 \leq A_i \leq 10^9$;

Giải thuật: Bạn chỉ cần lặp lại tất cả các phần tử của mảng và chọn những phần tử có giá trị lớn hơn hoặc bằng 0. Người ta chỉ cần tránh những lỗi ngớ ngẩn và xem xét các trường hợp góc. Một trường hợp góc như vậy là khi tất cả các số của mảng là âm. Trong trường hợp đó, bạn cần xuất số lượng tối đa của mảng.

Bài tập 8: Tên riêng Manna

Manna vô cùng thất vọng khi phát hiện ra rằng không ai trong trường đại học của mình biết tên của mình. Ngay cả bạn cùng lớp của anh ấy chỉ gọi anh ấy bằng tên cuối cùng của anh ấy. Thất vọng, anh quyết định làm cho các sinh viên đại học của mình biết tên của mình bằng cách buộc họ phải giải quyết câu hỏi này.

Bạn được cung cấp một chuỗi dài làm đầu vào trong mỗi testcase, chứa bất kỳ ký tự ASCII nào. Nhiệm vụ của bạn là tìm ra số lần SUVO và SUVOJIT xuất hiện trong đó.

Lưu ý: Vấn đề này phải được giải quyết chỉ bằng ngôn ngữ C.

Dữ liệu vào: Cho trong tệp văn bản FNAME.INP gồm

Dòng đầu tiên chứa số lượng testcase, T. Tiếp theo, T dòng theo sau mỗi dòng chứa một chuỗi dài S.

Dữ liệu ra: Ghi ra tệp văn bản FNAME.OUT gồm

Đối với mỗi chuỗi dài S, hiển thị số không. Lần SUVO và SUVOJIT xuất hiện trong đó.

Ví dụ:

FNAME.INP	FNAME.OUT
3	SUVO = 0, SUVOJIT = 1
SUVOJITSU	SUVO = 1, SUVOJIT = 0
651SUVOMN	SUVO = 1, SUVOJIT = 1
\$\$\$\$\$SUVOSUVOJIT\$\$\$\$\$	

Ràng buộc:

- $1 \leq T \leq 100$;
- $1 \leq \text{Length of each string} \leq 150$.

Giải thuật: Vậy chỉ có ngôn ngữ C hả? Cảm thấy như một thượng cổ chưa? Chỉ khi có Python và chúng ta mới có thể giải quyết nó bằng một chút phép thuật ngược. Mã này có sẵn ở đây.

```
for i in range(int(raw_input())):
    name = raw_input()
    suvo = name.count('SUVO')
    suvojit = name.count('SUVOJIT')
    print "SUVO = %d, SUVOJIT = %d"%(suvo-suvojit,suvojit)
```

Bài tập 9: Giao dịch Square

Square Inc. xử lý hàng ngàn giao dịch hàng ngày lên tới hàng triệu đô la. Họ cũng có một mục tiêu hàng ngày mà họ phải đạt được. Đưa ra một danh sách các giao dịch được thực hiện bởi Square Inc. và mục tiêu hàng ngày nhiệm vụ của bạn là xác định xem giao dịch nào mà Square đạt được như vậy.

Dữ liệu vào: Cho trong tệp văn bản **SQUARE.INP** gồm

Dòng đầu tiên chứa T, số lượng giao dịch được thực hiện bởi Square trong một ngày.

Dòng sau chứa số nguyên T, giá trị của mỗi giao dịch.

Dòng tiếp theo chứa Q, không có truy vấn.

Các dòng Q tiếp theo chứa một số nguyên, mỗi số đại diện cho mục tiêu hàng ngày.

Dữ liệu ra: Ghi ra tệp văn bản **SQUARE.INP** - đối với mỗi truy vấn, hãy in số giao dịch trong đó đạt được giới hạn hàng ngày hoặc -1 nếu không thể đạt được mục tiêu.

Ví dụ:

SQUARE.INP	SQUARE.OUT
5	3
1 2 1 3 4	2
3	5
4	
2	
10	

Ràng buộc (*Constraints*):

- $1 \leq T \leq 100000$;
- $1 \leq A_i \leq 1000$;
- $1 \leq \text{Target} \leq 10^9$;
- $1 \leq Q \leq 100000$.

Bài tập 10: Cửa hàng đặc biệt

Creatnx bây giờ muốn trang trí ngôi nhà của mình bằng những chậu hoa. Anh dự định mua chính xác N. Anh ta chỉ có thể mua chúng từ cửa hàng của Triracle. Chỉ có hai loại chậu hoa có sẵn trong cửa hàng đó. Quán rất lạ. Nếu bạn mua chậu hoa X loại 1 thì bạn phải trả $A \times X^2$ và $B \times Y^2$

nếu bạn mua chậu hoa Y loại 2. Hãy giúp Creatnx mua chính xác N chậu hoa để giảm thiểu số tiền anh ấy trả.

Dữ liệu vào: Cho trong tệp văn bản gồm

- Dòng đầu tiên chứa số nguyên T biểu thị số lượng trường hợp thử nghiệm.
- Mỗi trường hợp thử nghiệm được mô tả trong một dòng duy nhất chứa ba số nguyên được phân tách bằng dấu cách N, A, B.

Dữ liệu ra: Ghi ra tệp văn bản - Đối với mỗi trường hợp kiểm tra, in một dòng duy nhất có chứa câu trả lời.

Ví dụ:

SHOP.INP	SHOP.OUT
2	17
5 1 2	134
10 2 4	

Ràng buộc (Constraints):

- $1 \leq T \leq 10^5$
- $1 \leq N, A, B \leq 10^5$

Giải thích:

- Truy vấn 1: chúng ta phải mua chính xác 5 chậu. Có sáu lựa chọn có thể:
- Mua 0 nôi loại thứ nhất, 5 nôi loại thứ hai. Chi phí là: $1 \times 0^2 + 2 \times 5^2 = 50$.
- Mua 1 nôi loại thứ nhất, 4 nôi loại thứ hai. Chi phí là: $1 \times 1^2 + 2 \times 4^2 = 33$.
- Mua 2 chậu loại thứ nhất, 3 chậu loại thứ hai. Chi phí là: $1 \times 2^2 + 2 \times 3^2 = 22$.
- Mua 3 chậu loại thứ nhất, 2 chậu loại thứ hai. Chi phí là: $1 \times 3^2 + 2 \times 2^2 = 17$.
- Mua 4 nôi loại thứ nhất, 1 nôi loại thứ hai. Chi phí là: $1 \times 4^2 + 2 \times 1^2 = 18$.
- Mua 5 nôi loại thứ nhất, 0 nôi loại thứ hai. Chi phí là: $1 \times 5^2 + 2 \times 0^2 = 25$.

Vì vậy, chi phí tối ưu là 17.

Giải thuật:

Chúng ta hãy giải thích vấn đề trong đại số:

Tìm mức tối thiểu của hàm $f(x, y) = ax^2 + by^2$ chủ đề: $x + y = n$ và x, y là các số nguyên không âm.

Hàm có thể được viết lại dưới dạng $g(x) = ax^2 + b(n - x)^2$. Người ta có thể dễ dàng xác minh rằng: $g(x) = (a + b)(x - bn/(a + b))^2 + c$, trong đó c là hằng số chỉ phụ thuộc vào a, b, n .

Vì vậy, chúng ta cần tối thiểu hóa $(x - bn/(a + b))^2$ chủ đề: x là một số nguyên không âm giữa 0 và n . Rõ ràng, hàm này là tối thiểu khi x là số nguyên gần nhất với $bn/(a + b)$. Người ta có thể sử dụng một hàm thích 'round' trong C / C++: $x = \text{round}(bn/(a + b))$, sau đó $y = n - x$.

Độ phức tạp thời gian: $O(1)$ cho mỗi truy vấn.

Bài tập 11: Kỳ nghỉ

Đó là một kỳ nghỉ lễ cho tất cả học sinh trên toàn thế giới! Thật không may, Mahamba đang bận rộn chuẩn bị cho kỳ thi Olympic quốc tế về tin học, sẽ được tổ chức tại Tehran, Iran. Bây giờ anh ấy đang đối mặt với một thử thách mới từ giáo viên Aceka của mình, và nó diễn ra như thế này:

Bạn có một chuỗi x có độ dài N , bao gồm các chữ cái tiếng Anh nhỏ. Bạn phải tìm số chỉ số a, b, c và d , sao cho $1 \leq a < b < c < d \leq N$ và $x_a == x_c$, cũng như $x_b == x_d$.

Anh ấy đang bối rối và chắc chắn cần sự giúp đỡ. Vì vậy, bạn, lập trình viên giỏi nhất ở Lalalandia, đã quyết định giúp anh ta một tay!

Dữ liệu vào: Cho trong tệp văn bản

Dòng đầu tiên chứa số N - chiều dài của chuỗi x . Dòng thứ hai chứa chuỗi x chính nó.

Dữ liệu ra: Ghi ra tệp văn bản

Dòng đầu tiên và duy nhất nên chứa câu trả lời cho vấn đề.

Ví dụ:

HSEASON.INP	HSEASON.OUT
5 Ababa	2

Giải thích: Chỉ có 2 biến thể: $(a = 1, b = 2, c = 3, d = 4)$ và $(a = 2, b = 3, c = 4, d = 5)$

Ràng buộc:

- $1 \leq N \leq 2000$
- Chuỗi x chỉ chứa các chữ cái tiếng Anh nhỏ.

Giải thuật: Đầu tiên, với mỗi ký tự c cho mỗi $0 \leq i \leq n$ tính số lần xuất hiện của c trong số i ký tự đầu tiên của x . Bây giờ chúng ta có thể tìm thấy số lần xuất hiện của bất kỳ ký tự nào trong bất kỳ chuỗi con nào trong thời gian $O(1)$. Sau đó lặp qua các vị trí b và d và nếu $x_b = x_d$, tìm số vị trí a và c sao cho $x_a = x_c$. Số này có thể được tìm thấy trong thời gian $O(\alpha)$, trong đó α là kích thước của bảng chữ cái bằng cách lặp lại giá trị của x_a .

Bài tập 11: Cảnh sát và tên trộm

Bạn được cung cấp một lưới có kích thước $N \times N$ có các thông số kỹ thuật sau: Mỗi ô trong lưới chứa một cảnh sát hoặc một tên trộm.

- Một cảnh sát chỉ có thể bắt được một tên trộm nếu cả hai người ở cùng một hàng.
- Mỗi cảnh sát chỉ có thể bắt được một tên trộm.
- Một cảnh sát không thể bắt được một tên trộm cách xa cảnh sát hơn K đơn vị.

Viết chương trình tìm số kẻ trộm tối đa có thể bắt được trong lưới.

Dữ liệu vào: Cho trong tệp văn bản

- Dòng đầu tiên: T (số trường hợp kiểm tra)
- Đối với mỗi trường hợp thử nghiệm
- Dòng đầu tiên: Hai số nguyên cách nhau N và K
- N dòng tiếp theo: N ký tự được phân tách bằng dấu cách (biểu thị mỗi ô trong lưới)

Dữ liệu ra: Đối với mỗi trường hợp thử nghiệm, in số lượng kẻ trộm tối đa có thể bị bắt trong lưới.

Ví dụ:

THIEVES.INP	THIEVES.OUT
1 3 1 P T P T P T T T P	3

Giải thích: Tổng số kẻ trộm = 5

- Số kẻ trộm có thể tiếp cận được bởi cảnh sát ở Hàng 1 = 1
- Số kẻ trộm có thể tiếp cận được bởi cảnh sát ở Hàng 2 = 2
- Số kẻ trộm có thể tiếp cận được bởi cảnh sát ở Hàng 3 = 1

Tuy nhiên, một cảnh sát có thể bắt được tối đa 1 tên trộm. Do đó, trong Hàng 2, chỉ có 1 tên trộm là có thể bắt được. Do đó, 3 tên trộm có thể bị bắt.

Ràng buộc:

- $1 \leq T \leq 10$;
- $1 \leq N \leq 1000$;
- $1 \leq K \leq N * N$;