### Bài 1: Tổng bằng x

Cho hai dãy số nguyên a1, a2, a3, …., am và b1, b2, b3, …, bn và một số nguyên x. Hãy đếm xem có bao nhiêu cặp (i,j) thỏa mãn ai+bj=x.

Input: File sumx.inp:

* Dòng đầu tiên ghi ba số nguyên dương m, n, x (1≤m,n≤100000).
* Dòng thứ hai ghi các số nguyên a1, a2, a3, .., am (│ai│≤109).
* Dòng thứ ba ghi các số nguyên b1, b2, …, bn ((│bj│≤109).

Output: File sumx. out một số nguyên duy nhất là số cặp (i,j) tìm được

Example:

|  |  |
| --- | --- |
| SUMX.INP | SUMX.OUT |
| 4 5 5  3 1 4 2  1 6 4 3 4 | 4 |

**Bài 2: Trắc nghiệm tâm lí**

Trắc nghiệm tâm lí là phương pháp thông dụng để có thể đoán nhận được tính cách của mỗi người trong cuộc sống và cũng là một trò chơi khá phổ biến trên truyền hình. Trong một tò chơi như vậy được phát trên kênh NTTV. Trước tiên, ban tổ chức phát cho mỗi khán giả ngồi xem trực tiếp một phiếu thăm dò trong đó có các câu hỏi trắc nghiệm. Tất cả các phương án trả lời đều có điểm và mỗi người sau khi trả lời xong sẽ được tổng điểm là một số nguyên dương. Có m người tham gia cuộc chơi trên sân khấu. Với người chơi thứ i , sau khi nghe speaker đọc các câu hỏi trắc nghiệm sẽ đưa ra hai số nguyên si và fi với ý nghĩa rằng những khán giả có tổng điểm nằm trong đoạn [si, fi] sẽ là những người có tính cách phù hợp với mình nhất.

Viết chương trình tính xem mỗi người chơi sẽ tìm thấy bao nhiêu khán giả có tính cách phù hợp với mình nhất.

Input: File prefer.inp:

* Dòng đầu tiên chứa số nguyên n (0<n<=105) là số khán giả.
* Dòng thứ hai chứa n số nguyên dương a1, a2, ..,an (ai<=109) là tổng điểm của mỗi khán giả.
* Dòng thứ 3 ghi số nguyên m (1<m<=105) là số người chơi
* M dòng tiếp theo, dòng thứ i ghi hai số nguyên si, fi (1<=si<=fi<=109) là khoảng điểm của những người có tính cách phù hợp nhất với người i nhất.

Output: File prefer.out gồm m dòng, dòng thứ i ghi số lượng khán giả có tính cách phù hợp với người thứ i nhất.

Ví dụ:

|  |  |
| --- | --- |
| PREFER | PREFER |
| 5  7 2 4 5 3  2  1 4  3 10 | 3  4 |

**Bài 3: ESEQ**

Cho dãy số nguyên*A* gồm*n* phần tử*a1, a2, ..., an*. Hãy đểm số cặp chỉ số*i*,*j* thỏa mãn:

=Với 1≤*i*<*j*≤*n*.

Input*:* Vào từ file ESEQ.INP có dạng:

* Dòng đầu ghi số nguyên dương*n* (2≤*n*≤105)
* Dòng tiếp theo ghi*n* số nguyên*a1, a2, ..., an* (*ai* <= 109 ) các số cách nhau bởi dấu cách

Output*:* Ghi ra file văn bản ESEQ.OUT một số duy nhất là số cặp tìm được

Ví dụ:

|  |  |
| --- | --- |
| eseq.inp | eseq.out |
| 3  1 0 1 | 3 |

Thuật toán: Việc đầu tiên ta đi tìm mảng tổng tiến tố s[i]=a[i]+s[i-1]

Sau đó để tính tổng trên đoạn i…j ta chỉ cần lấy s[j]-s[i-1]

Sau đó sắp xếp mảng s tăng dần. Trong quá trình sắp xếp mảng s dẫn đến chỉ số thay đổi do đó ta phải dùng một mảng thứ hai để lưu chỉ số trước khi sắp xếp. Dùng kiểu dữ liệu cặp để khi mảng s thay đổi thì kéo theo cả chỉ số luôn.

**Bài 4: Chạy đua**

Bờm được Cuội mời lên mặt trăng để tham quan cơ ngơi của mình. Đã lâu không gặp đôi bạn quyết định tổ chức chạy thi tranh giải. Do đã sống lâu trên mặt trăng nên Cuội luôn chạy nhanh gấp đôi Bờm (tuy vậy nhanh hơn nữa là điều không thể!). Biết ưu thế đó của chồng mình, Hằng Nga tổ chức chạy thi như sau:

Có *n* vị trí phân biệt được đánh dấu dọc theo một đường thẳng. Bờm xuất phát từ vị trí X và chạy về bên phải còn Cuội xuất phát từ vị trí Z chạy về phía bên trái. Tại một vị trí Y nào đó nằm giữa hai vị trí trên Hằng Nga đặt một phần thường và vì là vợ của Cuội nên tất nhiên cô ta sẽ chọn vị trí Y sao cho Bờm không thể tới Y trước Cuội. Tuy vậy, Cuội luôn muốn mình không xuất phát ở điểm ở gần đích hơn so với Bờm.

Hãy đếm xem có bao nhiêu cặp vị trí (X,Y,Z) với X<Y<Z thỏa mãn điều kiện trên?

Input: vào file running .inp

* Dòng đầu tiên ghi số nguyên N - số vị trí đánh dấu (N≤1000)
* N dòng tiếp theo, mỗi dòng chứa một số nguyên là một vị trí đánh dấu (có giá trị nằm trong khoảng [0...109])

Output: Ghi ra file running .out

Một dòng ghi một số nguyên duy nhất là số bộ (X, Y, Z) tìm được

*Example*

|  |  |
| --- | --- |
| running .inp | running .out |
| 5  3  1  10  7  4 | 4 |

Thuật toán: Không mất tính tổng quát ta sắp xếp lại dãy số theo thứ tự không giảm.Ver 1. Do Cuội luôn xuất phát ở vị trí không gần hơn đích so với Bờm nên cách đơn giản là duyệt các cặp số thoả mãn ay-ax<=az-ay (trong đó az là vị trí của Cuội, ax là vị trí của Bờm, ay là vị trí phần thưởng) và do cuội luôn chạy nhanh gấp đôi so với Bờm. nên az-ay<=2\*(ay-ax). Như vậy cứ đếm bao nhiêu cặp thoả mãn thì dem++.

Code chương trình :

sort(a=1,a+n+1);

dem=0;

for (int i=1; i<=n-2; i++)

for (int j=i+1; j<=n-1; j++)

for (int z=i+1; j<=n; j++)

if ( a[j]-a[i]<=a[z]-a[j] && (a[z]-a[j]<=2\*(a[j]-a[i])) dem++;

Với thuật toán này chỉ được 50% số điểm. Để được full điểm cần xử lí bớt vòng lặp for bên trong. Đặt d=aj-ai

Ta nhận thấy do aj-ai<=az-aj && az-aj<=2\*(aj-ai)

=> d<=az-aj<2\*d => d+aj<=az<=aj+2d

Như vậy ta sẽ dùng hai hàm tìm kiếm nhị phân đề tìm các vị trí thoả mãn

Dem=0;

For (int i=1; i<=n-1; i++)

For (int j=i+1; j<=n; j++) { d=a[j]-a[i];

ans+=upper\_bound(a+1,a+n+1,a[j]+2\*d)-lower\_bound(a+1,a+n+1,a[j]+d);}

cout<<ans;

**Bài 5: Giá trị lớn nhất**

Bạn được cho một dãy số nguyên . Hãy tìm giá trị lớn nhất của (phần dư của phép chia số nguyên cho ) với và .

*Input:vào file* MAXVAL.INP

* Dòng đầu tiên chứa số nguyên dương - độ dài của dãy (
* Dòng thứ hai chứa số nguyên cách nhau bởi dấu trống

*Output:* Ghi ra file MAXVAL.OUT Kết quả tìm được

*Example:*

|  |  |
| --- | --- |
| MAXVAL.INP | MAXVAL.OUT |
| **3**  **2 4 5** | **1** |

*Ghi chú: 50% số test có*

Thuật toán: Để tìm giá trị lớn nhất của phép chia dư a[i] mod a[j] ta dễ dàng tìm được khi sử dụng 2 vòng lặp for.

for (int i=1; i<=n; i++)

for (int j=1; j<=n; j++) {

if (a[i]>=a[j]) r=a[i] %a[j];

res=max(res,r); }

Với cách này chỉ được 50% số điểm. Để cải tiến chương trình ta nhận thấy nếu các giá trị trong mảng a được sắp xếp theo thứ tự: a1<a2<….<an

Thì ta chỉ cần xét những giá trị gần 2\*a[i], 3\*a[i], …., ka[i] sẽ luôn cho giá trị chia dư lớn nhất. Như vây tại sử dụng tìm kiếm nhị phần tìm vị trí cuối cùng<=2\*a[i] khi đó đây là phần tử luôn cho giá trị lớn nhất khi chia dư cho a[i]

### Bài 6: Hẹn gặp

(*Nguồn: Đề thi HSG cấp Tỉnh Thanh Hóa năm 2016-2017*)

Thành phố Gloaming (*Hoàng hôn*) nổi tiếng với đường dẫn vào công viên thành phố. Các bức tượng tuyệt đẹp theo chủ đề thần thoại Hy lạp – La mã đặt dọc theo con đường thẳng có một sức hút không cưỡng được với mọi khách du lịch. Còn khi những tia nắng cuối cùng trong ngày miễn cưỡng rời khỏi bầu trời thì sương mù dày đặc, như một tấm voan trắng mềm mại từ từ rũ xuống. Bây giờ đứng cách quá ***r*** mét là đã không nhìn thấy mặt nhau và các bức tượng trở thành nơi lý tưởng cho các đôi nam nữ thanh niên hẹn hò.

James Bond cần gặp gấp 2 điệp viên nội tuyến của mình để nhận các mật báo khẩn. Không muốn 2 người này nhìn thấy nhau, Bond hẹn gặp mỗi người ở một bức tượng sao cho khoảng cách giữa chúng lớn hơn ***r***. Trên đường có ***n*** bức tượng, bức tượng thứ ***i*** ở vị trí cách đầu con đường ***di*** mét ***i*** = 1 ÷ ***n***, 1 ≤ ***d1***< ***d2*** < . . .< ***dn*** ≤ 109.

***Yêu cầu:*** Hãy xác định James Bond có bao nhiêu cách chọn địa điểm.

***Dữ liệu vào:*** Vào từ file văn bản HENGAP.INP:

- Dòng đầu tiên chứa 2 số nguyên ***n*** và ***r*** (1 ≤ ***n*** ≤ 3×105, 1 ≤ ***r*** ≤ 109).

- Dòng thứ 2 chứa ***n*** số nguyên ***d1***, ***d2***, . . ., ***dn***.

***Kết quả:*** Đưa ra file văn bản HENGAP.OUT một số nguyên là số cách chọn địa điểm tìm được.

***Ví dụ:***

|  |  |
| --- | --- |
| HENGAP.INP | HENGAP.OUT |
| 4 4  1 3 5 8 | 2 |

*Rằng buộc:*

* Có ½ số test tương ứng với ½ số điểm có ***n*** ≤ 104
* Có ½ số test tương ứng với ½ số điểm có 104 < ***n*** ≤ 3×105

**Ý tưởng đề xuất:**

- Dùng một vòng lặp For – do để duyệt i từ 1 đến n-1, trong đó với mỗi vị trí i thì ta tìm kiếm nhị phân để tìm địa điểm j xa nhất thỏa mãn r+di < dj.

### Bài 7. Đếm cặp số

Số N được gọi là một số lũy thừa của 2 nếu tồn tại số x sao cho 2x = N.

Cho dãy N số nguyên dương a1, a2,..., aN. Đếm số lượng các cặp số i,j (1 i < j N) sao cho ai + aj là một số lũy thừa của 2.

Dữ liệu vào: từ tệp văn bản DEMCS.INP có cấu trúc như sau:

- Dòng đầu ghi số nguyên dương N là số lượng các số của dãy;

- Dòng thứ hai ghi N số nguyên dương a1, a2,..., aN cách nhau một dấu cách.

Kết quả: ghi ra tệp văn bản DEMCS.OUT số lượng cặp đếm được.

Ví dụ:

|  |  |
| --- | --- |
| DEMCS.INP | DEMCS.OUT |
| 5  1 2 3 7 1 | 5 |

Ràng buộc:

- Có 50% test n 1.000 và ai 1.000.000 tương ứng 50% số điểm;

- Có 30% test n < 100.000 và ai 1.000.000 tương ứng 30% số điểm;

- Có 20% test n 100.000 và ai 1.000.000.000 tương ứng 20% số điểm.

**Bài 8: seagame** *(Bài 3.18 sách chuyên Tin quyển 1- VOI2008)*

Hai bạn học sinh lúc ngồi nhàn rỗi nghĩ ra trò chơi sau đây, Mỗi bạn chọn trước một dãy số gồm n số nguyên, Giải sử dãy số mà bạn thư nhất chọn là b1, b2, ..,bn, còn dãy số mà bạn chọn thứ 2 là c1, c2, …,cn. Mỗi lượt chơi, mỗi bạn đưa ra một số hạng trong dãy số của mình. Nếu bạn thứ nhất chọn số bi (0<i≤n), bạn thứ 2 chọn số cj (0<j≤n)thì giá trị của lượt chơi sẽ là |bi+cj|. Hãy tìm giá trị nhỏ nhất của một lượt chơi trong số các lượt chơi có thể.

File input sgame.inp gồm có:

* Dòng đầu tiên chứa số nguyên dương n (1<n<=105)
* Dòng thứ hai chúa dãy số nguyên b1, b2, …,bn (|bi|≤109 ,i=1,…,n);
* Dòng thứ ba chứa dãy số nguyên c1, c2, …,cn (|cj|≤109, j=1,2,…,n);

File ouput sgame.out một số nguyên duy nhất là giá trị nhỏ nhất tìm được.

Ví dụ:

|  |  |
| --- | --- |
| **sgame.inp** | **sgame.out** |
| 2  1 -2  2 3 | 0 |

**Ý tưởng đề xuất**:

- Nhận xét thấy rằng bi+cj càng nhỏ khi cj càng gần –bi như vậy việc đầu tiên ta sẽ sắp xếp lại mảng c1≤c2≤c3≤…≤cn.

- Sau đó dùng hàm lower\_bound() để tìm vị trí xuất hiện đầu tiên của –b[i] trong mảng c mà –b[i]<=C do vậy nếu k=lower\_bound(c+1,c+n+1,–b[i])-c thì , ck-1<-bi≤ck

### Bài 9: Nấu ăn

Nghệ nhân nấu ăn Tư Mập có thể sử dụng hệ thống gồm n bếp điện để thực hiện nấu món ăn khiến ông được vinh danh, đó là món “ga tô hải sản”. Thời gian để thực hiện nấu một suất ăn như vậy trên các bếp điện tương ứng là t1,t2,…,tn giây.

**Yêu cầu:**

Cho biết s là số lượng thực khách cần phục vụ, hãy xác định thời gian tối thiểu cần thiết để nghệ nhân có thể nấu xong s suất ăn trên hệ thống bếp điện của khách sạn. Để nấu mỗi suất ăn chỉ được sử dụng một bếp.

**Dữ liệu:**

Vào từ file văn bản CAKES.INP

Dòng đầu tiên chứa số lượng suất ăn s (0<s<1015) và số lượng bếp điện n (0<n<20)

Dòng thứ hai chứa n số nguyên dương t1,t2,…,tn mỗi số nhỏ hơn 500.

**Kết quả:**

Ghi ra file văn bản CAKES.OUT duy nhất một số nguyên là thời gian tối thiểu tìm được tính bằng giây.

**Ví dụ:**

|  |  |
| --- | --- |
| CAKES.INP | CAKES.OUT |
| 3 3  10 70 60 | 30 |

### Bài 10: Tìm giữa

*(Nguồn: Đề thi HSG lớp 12 thành phố Hà Nội 2020-2021)*

Cho hai số nguyên dương L và R.

**Yêu cầu**: Tìm số nguyên dương M (L<=M<R) để chênh lệch giữa tổng các số nguyên liên tiếp từ L đến M và tổng các số nguyên liên tiếp từ M+1 đến R là nhỏ nhất.

**Dữ liệu**: Vào từ tệp văn bản BÀI 1.INP

- Gồm hai số nguyên dương L và R (L<R<109)

**Kết quả**: Ghi vào tệp văn bản BÀI 1.OUT

- Gồm một số nguyên duy nhất là số M thỏa mãn.

Ví dụ:

|  |  |  |
| --- | --- | --- |
| **BÀI 1.INP** | **BÀI 1.OUT** | **Giải thích** |
| 2 7 | 5 | Tổng từ 2 đến 5 là: 14  Tổng từ 6 đến 7 là 13  Chênh lệch là: 1 |

**Ý tưởng đề xuất:**

- Dựa vào miền kết quả theo đề bài, ta dùng phương pháp chặt nhị phân

- Đặt d=L, c=R

- Với giá trị mid = (L+R)/2

- Tính tổng từ d đến mid đặt là first\_sum, tổng từ mid +1 đến c đặt là last\_sum

- Cập nhật độ chêch lệch nhỏ nhất giữa first\_sum và las\_sum

- Nếu first\_sum < las\_sum thì điều chỉnh mid tăng lên tức là L=mid +1, ngược lại điều chỉnh mid giảm xuống tức là R=mid-1

### Bài 11: Trung bình lớn nhât

*(Nguồn: Đề thi HSG lớp 12 Tỉnh Hưng Yên năm 2020 -2021)*

Cho số nguyên ương n và dãy số A1, A2,….An. Gọi đoạn con [u,v] của dãy là các phần tử liên tiếp Au, Au+1, Au+2,…., Av (u<= v). Dễ thấy đoạn con [u,v] có độ dài là v-u+1 và giá trị trung bình là: 

**Yêu cầu**: Cho số nguyên k, hãy xác định đoạn con có độ dài không nhỏ hơn k có giá trị trung bình lớn nhất.

**Dữ liệu vào**: file văn bản AVERAGE. INP

- Dòng đầu chứa 2 số nguyên dương n,k

- Dòng thứ 2 chứa n số nguyên A1, A2,….An (Ai <= 109, 1 <=i<=n), các số được phân tách nhau bởi dấu cách

**Dữ liệu ra**: Đưa ra file văn bản AVERAGE.OUT một số thực duy nhất là giá trị trung bình của đoạn con tìm được. Kết quả đưa ra lấy 3 chữa số thập phân sau dấu phẩy

Ví dụ

|  |  |
| --- | --- |
| **AVERAGE. INP** | **AVERAGE.OUT** |
| 4 2  17 0 14 1 | 10.333 |
| 5 1  2 8 -1 4 5 | 8.000 |

Giải thích

- Trong ví dụ 1, đoạn con có giá trị trung bình lớn nhất thỏa mãn là đoạn [1, 3] với các giá trị 17, 0, 14

- Trong ví dụ 2, đoạn con có giá trị trung bình lớn nhất thỏa mãn là đoạn [2, 2] với giá trị là 8

**Ý tưởng đề xuất**

**Cách 1**:

- Tính mảng tổng tiền tố: pref\_sum[i]= a[1] + a[2] +….+ a[i] = pref\_sum[i-1] + a[i]

- Xét cặp (i,j) với i<= j sao chp j-i +1 >=k

- Kết quả là: res= max (res, (pref\_sum[i]- pref\_sum[i-1])/(j – i+1))

- Độ phức tạp O(n2)

**Cách 2:**

Thực hiện phương pháp chặt nhị phân theo kết quả.

- Ta biết cần tìm giá trị lớn nhất res thỏa mãn

>= res tương đương với (a[i]+… +a[j] ) >= res\*(j-i+1) Suy ra ta có: (a[i]- res) +…..+ (a[j] - res) >=0

- Thực hiện chặt nhị phân để tìm kết quả res. Với mỗi giá trị mid, ta giảm mỗi phần tử của dãy a[ ] đi mid đơn vị theo công thức trên, sau đó kiểm tra trong dãy có đoạn con liên tiếp nào có tổng dương hay không? Nếu có thì ghi nhận giá trị res = mid và tăng giá trị mid, ngược lại giảm giá trị mid.

- Độ phức tạp của thuật toán O(nlogn)

### Bài 12: Tam giác (Nguồn: Đề Thi HSG Tỉnh Thanh Hóa 2020-2021)

Hiền có N que tính, mỗi que có độ dài là một số nguyên dương và đôi một khác nhau. Hiền muốn tạo ra các tam giác bằng cách ghép ba que tính với nhau, độ dài mỗi cạch của tam giác là độ dài của một que tính.

**Yêu cầu**: Giúp Hiền tính xem tạo được bao nhiêu tam giác nhọn, tam giác vuông và tam giác tù khác nhau từ các que tính này. Hai tam giác được gọi là khác nhau nếu có ít nhất một que tính khác nhau

**Dữ liệu vào**: Đọc từ tệp CAU5.INP có cấu trúc như sau:

- Dòng đầu là số lượng que tính N (3<=N<=5000)

- Dòng thứ hai gồm N số nguyên dương không lớn hơn 104 là chiều dài của N que tính.

**Dữ liệu ra**: Ghi ra tệp CAU5.OUT gồm 3 số nguyên lần lượt là số lượng tam giác nhọn, tam giác vuông và tam giác tù tạo được

|  |  |
| --- | --- |
| **CAU5.INP** | **CAU5.OUT** |
| 6  2 12 9 10 3 15 | 2 1 4 |

**Giới hạn:**

- Có 25% test 3<=N<=300

- Có 75 % test còn lại không có ràng buộc gì

**Ý tưởng thuật toán đề xuất**

Để 3 số nguyên a,b,c tạo thành 3 cạnh của tam giác thì a,b,c phải thỏa mãn biểu thức:

Điều kiện để 3 số nguyên a,b,c tạo thành 3 cạnh của:

- Tam giác nhọn: a2 + b2 > c2

- Tam giác vuông: a2 + b2 = c2

- Tam giác tù: a2 + b2 < c2

**Cách 1:**

- Duyệt 3 vòng for lồng nhau để tìm a,b,c tạo thành 3 cạnh của một tam giác, nhọn, vuông, tù theo điều kiện đã nêu.

- Với cách này độ phức tạp của thuật toán là O(n3). Vì vậy, chỉ đạt 25% số điểm khi N nhỏ.

**Cách 2:**

- Sắp xếp que tính tăng dần theo chiều dài của h[i]

- Dùng 2 vòng for lồng nhau để xác định cạnh a= h[i], b=h[j] (i<j) của tam giác

- Để xác định số lượng tam giác, ta sẽ xác định số lượng giá trị c.

+ Sử dụng phương pháp chặt nhị phân trên đoạn từ h[j+1] đến h[n] để xác định giá trị c như sau:

+ tìm vị trí p1 nhỏ nhất thỏa mãn

+ tìm vị trí p2 lớn nhất thỏa mãn h[p1] < a+b

- Để xác định số lượng giá trị c tạo thành tam giác nhọn, ta sử dụng tìm kiếm nhị phân trên đoạn từ h[p1] đến h[p2] để tìm vị trí p lớn nhất sao cho (h[p])2 < a2 +b2  
 suy ra số lượng tam giác nhọn sẽ là c1=x-p1+1

Tương tự ta tìm được số lượng tam giác tù là c2

Đối với tam giác vuông, tìm hai vị trí x1 nhỏ nhất và x2 lớn nhất sao cho (h[x1])2 = (h[x2])2 = a2 + b2. Suy ra số lượng tam giác vuông là x2= x1 +1.

Với cách này độ phức tạp của thuật toán là O(n2logn)

### **Bài 13.** TRIANGLE

Cho ba dãy số nguyên dương *A*, *B* và *C*.

Dãy *A*: *a*1, *a*2, .., *an*

Dãy *B*: *b*1, *b*2, .., *bn*

Dãy *C*: *c*1, *c*2, .., *cn*

Xét ba loại câu hỏi sau:

1) Có bao nhiêu bộ 3 chỉ số (*i,j,k*) mà *ai*, *bj* và *ck* tương ứng là ba cạnh của một tam giác đều.

2) Có bao nhiêu bộ 3 chỉ số (*i,j,k*) mà *ai*, *bj* và *ck* tương ứng là ba cạnh của một tam giác cân (tam giác đều cũng được tính là tam giác cân)

3) Có bao nhiêu bộ 3 chỉ số (*i,j,k*) mà *ai*, *bj* và *ck* tương ứng là ba cạnh của một tam giác.

**Yêu cầu:** Cho ba dãy số *A*, *B*, *C* và loại câu hỏi loại *w* (*w* bằng 1, 2 hoặc 3 tương ứng với câu hỏi loại 1, 2 hoặc 3), hãy tính số lượng bộ 3 chỉ số (*i,j,k*) cho câu hỏi loại *w*.

**Input**

Dòng đầu tiên chứa *Q* (*Q* ≤ 10) là số lượng bộ dữ liệu. Tiếp đến là *Q* nhóm dòng, mỗi nhóm cho thông tin về một bộ dữ liệu theo khuôn dạng sau đây:

• Dòng thứ nhất chứa hai số nguyên dương *n, w;*

• Dòng thứ hai ghi *n* số nguyên dương *a*1, *a*2, .., *an* (*ai* ≤ 109);

• Dòng thứ ba ghi *n* số nguyên dương *b*1, *b*2, .., *bn* (*bi* ≤ 109);

• Dòng thứ tư ghi *n* số nguyên dương *c*1, *c*2, .., *cn* (*ci* ≤ 109).

**Output**

Gồm *Q* dòng, mỗi dòng chứa một số nguyên là số lượng bộ 3 chỉ số (*i,j,k*) cho câu hỏi loại *w* tương ứng với bộ dữ liệu vào.

|  |  |
| --- | --- |
| TRIANGLE.INP | TRIANGLE.OUT |
| 2 2 2 10 30 30 10 10 20 2 3 20 30 30 10 10 20 | 3 4 |

**Subtask 1:** *n* ≤ 100; [20 tests]

**Subtask 2:** *n* ≤ 1000; *w =* 1; [20 tests]

**Subtask 3:** *n* ≤ 1000; *w =* 2; [20 tests]

**Subtask 4:** *n* ≤ 1000; *w =* 3. [20 tests]

**Subtask 5:** *n* ≤ 1000; [20 tests]

### **Bài 14: JUMP** (Nguồn Trại hè Hùng Vương lớp 11 năm 2014)

Cho dãy A gồm N số nguyên không âm A1, A2,…, AN. Một bước nhảy từ phần tử Ai đến phần tử Aj được gọi là bước nhảy xa nhất của dãy nếu thỏa mãn các điều kiện sau:

* 1 ≤ i < j ≤ N.
* Aj – Ai ≥ P.
* j – i lớn nhất

Khi đó j – i được gọi là độ dài bước nhảy xa nhất của dãy.

**Yêu cầu:** Tìm độ dài bước nhảy xa nhất của dãy A.

**Dữ liệu vào :** Từ tệp JUMP.INP có cấu trúc như sau:

* Dòng 1: Gồm hai số nguyên N và P (1 ≤ N ≤ 105; 0 ≤ P ≤ 109).
* Dòng 2: Gồm N số nguyên A1, A2,…, AN (0 ≤ Ai ≤ 109 với 1 ≤ i ≤ N). ( Các số cách nhau ít nhất 1 dấu cách )

**Kết quả :** Ghi vào tệp JUMP.OUT gồm một số nguyên dương duy nhất là độ dài của bước nhảy xa nhất của dãy (Nếu không có bước nhảy nào thỏa mãn thì ghi kết quả bằng 0).

**Ví dụ:**

|  |  |
| --- | --- |
| **JUMP.INP** | **JUMP.OUT** |
| 6 3  4 3 7 2 6 4 | 3 |

**Rằng buộc:**

*- Có 70% test ứng với N ≤ 5000.*

**Ý tưởng đề xuất:**

Gọi L[i] là giá trị nhỏ nhất của dãy A từ phần tử thứ 1 đến phần tử thứ i.  
L[1]:=A[1];  
L[i]:=Min(L[i-1],A[i]), i=2,3,..., N

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i | 1 | 2 | 3 | 4 | 5 | 6 |
| Ai | 4 | 3 | 7 | 2 | 6 | 4 |
| Li | 4 | 3 | 3 | 2 | 2 | 2 |

Với mỗi vị trí j = 2, 3 .., n ta tìm vị trí i ∈[1, j-1] nhỏ nhất sao cho a[j]-  
P≥L[i]. Khi đó ta được đoạn j-i thỏa mãn điều kiện bài toán, so sánh độ dài đoạn [i,j] với phương án tối ưu.

Vì 1 ≤ N ≤ 105 , Mảng L là mảng không tăng nên ta có thể áp dụng thuật toán tìm kiếm nhị phân trong đoạn [1, j-1] để tìm vị trí i nhỏ nhất sao cho a[j]-P≥L[i].

### Bài 15: Đi chơi (Đề thi HSG THCS cấp tỉnh Lào Cai năm 2021-2022)

Sau bao ngày học hành vất vả và dịch Covid-19 đã được kiểm soát, Tít cùng M người bạn của mình lên kế hoạch để đi chơi. Nhà của Tít và các bạn của Tít nằm trên cùng 1 con đường, các nhà được đánh vị trí từ 1 đến N, mỗi nhà cách nhau 1 mét. Nhà của Tít ở vị trí 1 và địa điểm vui chơi ở vị trí N. Nhà M người bạn của Tít ở các vị trí a1, a2,..., aM. Ngoài ra trên tuyến đường còn có P trạm xe buýt tại các vị trí b1, b2, ..., bP.

Từ nhà mình, Tít lần lượt đi đến nhà của các bạn mình theo kế hoạch. Tít có thể đi bằng taxi hoặc xe buýt. Với taxi, Tít có thể bắt từ bất kì vị trí nào, giá của taxi là T đồng/mét. Với xe buýt, Tít chỉ có thể bắt từ trạm này và đi đến một trạm khác, giá của xe buýt là B đồng/lượt không phân biệt khoảng cách. Do còn phải để dành tiền để đi chơi, Tít không thể lãng phí quá nhiều tiền cho việc đi lại. Bạn hãy giúp Tít tìm cách đi đón tất cả các bạn và đến điểm vui chơi với số tiền phải trả là ít nhất nhé!

**Yêu cầu:** Cho biết số nhà trên đường, các nhà phải đến đón, số trạm xe buýt và số tiền đi xe taxi, xe buýt, bạn hãy tìm cách đi cho Tít sao cho đến thăm đúng thứ tự các nhà và đến vị trí N với số tiền ít nhất.

**Dữ liệu vào:** Đọc dữ liệu vào từ tệp **BAI05.INP** có dạng:

- Dòng thứ nhất chứa các số nguyên N, M, P, T, B: Là vị trí điểm vui chơi, các nhà phải đón, số trạm xe buýt và số tiền đi taxi, xe buýt (1 ≤ N ≤ 109 | 0 ≤ M,P ≤ 105 | 1 ≤ T,B ≤ 104).

- Dòng thứ hai chứa M số nguyên là thứ tự các nhà phải đến, số thứ ai là vị trí của nhà thứ i (1 ≤ ai ≤ N). Dữ liệu cho đảm bảo không có 2 nhà trùng vị trí.

- Dòng cuối cùng chứa P số nguyên là vị trí các trạm xe buýt theo thứ tự tăng dần, số thứ bi là vị trí của trạm thứ i, mặc định có trạm ở vị trí 1 và N (1 ≤ bi ≤ N).

**Dữ liệu ra:** Ghi kết quả ra tệp **BAI05.OUT** có gồm 1 số nguyên duy nhất là số tiền ít nhất phải trả.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **BAI05.INP** | **BAI05.OUT** | **Giải thích** |
| 10 2 2 1000 2000  5 8  4 7 | 8000 | - Đầu tiên Tít đi xe buýt từ 1 đến 4.  - Sau đó đi taxi từ 4 đến 5, 5 đến 8 và 8 đến 10.  Tổng số tiền là 2000+1000+3000+2000=8000 đồng. |

**Ràng buộc:**

* **Subtask 1 (40%):** N,M,P ≤ 20.
* **Subtask 2 (40%):** N,M,P ≤ 100.
* **Subtask 3 (20%):** N ≤ 109; 0≤ M,P ≤ 105.

**Ý tưởng đề xuất:** Thực hiện tìm kiếm nhị phân để tìm ra điểm xe buýt gần nhất với từng địa điểm nhà bạn và lưu vào mảng TB.

* Để tìm chi phí rẻ nhất cho cách đi từ nhà thứ i đến nhà i+1 ta chỉ cần so sánh giá trị (khoảng cách giữa 2 nhà)\* số tiền đi taxi và (khoảng cách từ điểm 2 xe buýt gần nhất tới 2 ngôi nhà)\* số tiền đi taxi + số tiền đi 1 lượt xe buýt.
* Ví dụ như hình dưới: Ký hiệu B1, B2 là vị trí nhà các bạn của Hòa; N1, N2 là vị trí các điểm xe buýt gần nhất với 2 ngôi nhà thì ta so sánh (abs(B1-N1)+abs(B2-N2))\*Số tiền đi taxi +Số tiền cho mỗi lượt đi xe buýt và (B2-B1)\*Số tiền đi taxi

B1

B2