## Mục lục

## Bài 1. Số Fibo

Dãy số Fibonacci được Fibonacci, một nhà toán học người Ý, công bố vào năm 1202 trong cuốn sách Liber Abacci - Sách về toán đồ qua 2 bài toán: Bài toán con thỏ và bài toán số các "cụ tổ" của một ong đực. Dãy Fibonacci là dãy vô hạn các số tự nhiên bắt đầu bằng hai phần tử 0 và 1 hoặc 1 và 1, các phần tử sau đó được thiết lập theo quy tắc mỗi phần tử luôn bằng tổng hai phần tử trước nó. Công thức truy hồi của dãy Fibonacci là:

$$\begin{cases} ¿F_1 = F_2 = 1 \\ ¿F_n = F_{n-1} + F_{n-2} (n \geq 3) \end{cases}$$

**Yêu cầu:** Cho Q câu hỏi, mỗi câu hỏi là một số N. Hãy viết chương trình tính số fibo thứ N.

**Giới hạn:** N , $Q \leq 10^5 Q \leq 10^5$

**Dữ liệu vào:**

- Dòng đầu tiên nhập vào số nguyên dương Q.
- Q dòng tiếp theo, mỗi dòng là một số nguyên dương N

**Kết quả:** In ra Q dòng, mỗi dòng là một số fibo tương ứng với truy vấn nhập vào. Vì dữ liệu bài này rất lớn, nên hãy in ra kết quả chia lấy dư $(10^9+7)$.

**Ví dụ:**

| Input | output |
|---|---|
| **5**<br>**5**<br>**2** | **5**<br>**1**<br>**2** |

| | |
|---|---|
| 3 | 8 |
| 6 | 1 |
| 1 | |

## Bài 2. Tổng tiền tố

Trong tin học, mọi người thường có thuật ngữ như sau. Tiền tố, hay còn gọi là prefix. Tổng tiền tố, còn gọi là prefixsum. Cho một dãy A có N phần tử. Người ta thường hiểu rằng, tổng tiền tố kết thúc tại i là tổng $A_1+A_2+A_3+...+A_i$.

**Yêu cầu:** Nhập vào dãy A có N phần tử. Cho Q câu hỏi, mỗi câu hỏi là một số nguyên dương k <= N. Hãy tính tổng tiền tố kết thúc tại k.

**Giới hạn:** N , $Q \leq 10^5$và $A_i \leq 10^9$

**Dữ liệu vào:**

- Dòng đầu tiên nhập vào số nguyên dương N.
- Dòng thứ hai, nhập vào N số nguyên, số thứ i là $A_i$
- Dòng thứ ba, nhập vào số nguyên dương Q là số truy vấn

• Q dòng tiếp theo, nhập vào số nguyên dương k.

**Kết quả:** In ra Q dòng, mỗi dòng là tổng tiền tố tương ứng với truy vấn nhập vào.

**Ví dụ:**

| input | output |
|---|---|
| 5<br>1 -5 6 3 2<br>3<br>5<br>2<br>3 | 7<br>-4<br>2 |

## Bài 3. Chênh lệch độ cao

Cho N tòa nhà, tòa nhà thứ i có độ cao là $H_i$ .

**Yêu cầu:** Hãy tìm 2 toà nhà i và j (i < j). Sao cho $H_j$ - $H_i$ là lớn nhất có thể.

**Giới hạn:** $2 \leq N \leq 10^5$và $1 \leq H_i \leq 10^9$

**Dữ liệu vào:**

- Dòng đầu tiên nhập vào số nguyên dương N.
- Dòng thứ hai, nhập vào N số nguyên, số thứ i là $H_i$, độ cao tòa nhà thứ i

**Kết quả:** In ra số nguyên duy nhất là kết quả của bài toán.

**Ví dụ:**

| input | output |
|---|---|
| 5<br>6 5 1 2 4 | 3 |

**Bài 4. Tổng đoạnCho dãy A có N phần tử. Một đoạn con (i, j) của dãy A là các phần tử liên tiếp $A_i$, $A_{i+1}$, ... $A_j$ mà i <= j. Tổng của đoạn con là $A_i + A_{i+1} + ... + A_j$.**

**Yêu cầu:** Cho Q truy vấn, mỗi truy gồm hai số nguyên dương L và R. Hãy tính tổng đoạn (L, R)

**Giới hạn:** N , $1 \le H_i \le 10^9$;$Q \le 10^5$ và$A_i \le 10^9$. (1 <= L <= R <= N)

**Dữ liệu vào:**

- Dòng đầu tiên nhập vào số nguyên dương N.
- Dòng thứ hai, nhập vào N số nguyên, số thứ i là $A_i$
- Dòng thứ ba, nhập vào số nguyên dương Q là số truy vấn
- Q dòng tiếp theo, nhập vào hai số nguyên dương L và R.

**Kết quả:** In ra Q dòng, mỗi dòng là tổng đoạn tương ứng với truy vấn nhập vào.

**Ví dụ:**

| input | output |
|---|---|
| 5<br>1 -5 6 3 2<br>3<br>2 4<br>3 3<br>3 5 | 4<br>6<br>11 |

**Bài 5. Đoạn con có tổng lớn nhất.**

Cho dãy A có N phần tử. Một đoạn con của dãy A là dãy các phần tử liên tiếp $A_i$, $A_{i+1}$, ... $A_j$ mà i <= j. Tổng của đoạn con là $A_i + A_{i+1} + ... + A_j$.

**Yêu cầu:** Hãy tìm đoạn con có tổng lớn nhất.

**Giới hạn:** N <= $10^5$ và $A_i$ <= $10^9$

**Dữ liệu vào:**

- Dòng đầu tiên nhập vào số nguyên dương N.
- Dòng thứ hai, nhập vào N số nguyên, số thứ i là $A_i$

**Kết quả:** In ra một số nguyên duy nhất là tổng lớn nhất tìm được. **Ví dụ:**

| input | output |
|---|---|
| 6<br>4 3 -5 6 -9 2 | 8 |

**Bài 6. Hai đoạn con có tổng lớn nhất.**

Cho dãy A có N phần tử. Một đoạn con (i, j) của dãy A là dãy các phần tử liên tiếp $A_i$, $A_{i+1}$, ... $A_j$ mà i <= j. Tổng của đoạn con là $A_i + A_{i+1} + ... + A_j$. Hãy tìm hai đoạn con không giao nhau có tổng cả hai đoạn là lớn nhất. Điều này có

nghĩa là ta phải tìm bộ 4 chỉ số x <= y < u <= v. Sao cho tổng đoạn (x, y) và (u, v) là lớn nhất.
**Yêu cầu:** Hãy tìm hai đoạn con có tổng lớn nhất.
**Giới hạn** $2 \le N \le 10^5$ và $A_i \le 10^9$
**Dữ liệu vào:**

- Dòng đầu tiên nhập vào số nguyên dương N.
- Dòng thứ hai, nhập vào N số nguyên, số thứ i là $A_i$

**Kết quả:** In ra một số nguyên duy nhất là tổng lớn nhất của hai đoạn tìm được.
**Ví dụ:**

| input | output |
|---|---|
| 5<br>10 -4 -3 9 -2 | 19 |

## Bài 7. Truy vấn tăng đoạn.

Cho dãy số nguyên gồm N phần tử $A_1, A_2, \dots, A_n$. Ban đầu, các phần tử trong dãy đều bằng 0, sau đó bạn cần thực hiện Q thao tác mà mỗi thao tác phụ thuộc vào bộ ba số (L, R, K) (1 < L < R < N, |K| < 100), nghĩa là tăng các phần tử $A_L, A_{L+1}, \dots A_R$ lên K đơn vị.
**Yêu cầu:** Hãy in ra dãy A sau khi thực hiện Q thao tác.
**Giới hạn:** $2 \le N \le 10^5$ và $Q \le 10^5$
**Dữ liệu vào:**

- Dòng đầu tiên nhập vào hai số nguyên dương N và Q
- Q dòng tiếp theo, mỗi dòng nhập bộ ba số (L, R, K)

**Kết quả:** In ra N số nguyên là dãy A sau khi thực hiện Q thao tác.
**Ví dụ:**

| input | output |
|---|---|
| 5 2<br>1 3 7<br>2 5 -3 | 7 4 4 -3 -3 |

## Bài 8. Tổng hình chữ nhật.

Cho bảng N x M ô gồm có N hàng, M cột. Ô ở hàng i, cột j có giá trị là Ay. Ô (1, 1) ở góc trái trên, và ô (N, M) ở góc phải dưới. Tổng của hình chữ nhật có ô trái trên là (x,y) và ô phải dưới là (u, v) là tổng tất cả $A_{ij}$ mà x <= i <= u và y <= j <= v.
**Yêu cầu:** Cho Q truy vấn, mỗi truy vấn có dạng (x,y,u,v). In ra tổng của hình chữ nhật có góc trái trên là (x, y) và góc phải dưới là (u, v).
**Giới hạn:** $N, M \le 10^3$ và $Aij \le 10^9$
**Dữ liệu vào:**

- Dòng đầu tiên nhập vào hai số nguyên dương N và M
- N dòng tiếp theo, mỗi dòng gồm M số nguyên, dòng thứ i và số thứ j là

$A_{ij}$

- Cho số nguyên dương Q
- Q dòng tiếp theo, mỗi dòng chứa 4 số nguyên x, y, u, v.

**Kết quả:** In ra Q dòng, mỗi dòng là tổng hình chữ nhật của truy vấn tương ứng.

**Ví dụ:**

| input | output |
|---|---|
| **3 4** | **31** |
| **2 -5 6 3** | **-2** |
| **1 9 7 -2** | **15** |
| **3 5 2 10** | |
| **3** | |
| **2 2 3 4** | |
| **2 4 2 4** | |
| **2 1 2 4** | |

## Bài 9. Truy vấn tăng bảng.

Cho bảng N x M ô gồm có N hàng và M cột. Ô ở hàng i, cột j gọi là ô (i, j). Ô (1, 1) ở góc trái trên, và ô (N, M) ở góc phải dưới. Ban đầu, tất cả ô đều có giá trị 0. Cho Q truy vấn, mỗi truy vấn có dạng (x, y, u, v, k) với |k| <= 1000, bạn phải tăng các ô trong hình chữ nhật có góc trái trên là (x, y) và góc phải dưới là (u, v) lên k đơn vị. Nghĩa là bạn phải tăng tất cả ô (i, j) mà x <= i <= u và y <= j <= v lên k đơn vị.

**Yêu cầu:** In ra bảng sau khi thực hiện Q truy vấn.

**Giới hạn:** $N, M \leq 10^3$ và $Q \leq 10^5$

**Dữ liệu vào:**

- Dòng đầu tiên nhập vào ba số nguyên dương N, M và Q
- Q dòng tiếp theo, mỗi dòng chứa 5 số nguyên x, y, u, v, k

**Kết quả:** In ra Q dòng, mỗi dòng là tổng hình chữ nhật của truy vấn tương ứng.

**Ví dụ:**

| input | output |
|---|---|
| **3 4 3** | **2 2 2 0** |
| **2 2 3 4 4** | **1 5 5 3** |
| **2 1 2 4 -1** | **0 4 4 4** |
| **1 1 2 3 2** | |

## Bài 10. Khối lập phương lớn nhất

Khối lập phương là một khối đa diện đều ba chiều có 6 mặt là 6 hình vuông bằng nhau. Cho một khối lập phương kích thước N chia làm $N^3$ khối lập phương

đơn vị. Mỗi khối lập phương đơn vị chứa 1 số nguyên.
**Yêu cầu:** Bạn hãy tìm một khối lập phương con là tập hợp các khối lập phương đơn vị liền kề sao cho tổng các số trong khối lập phương con đó là lớn nhất.
**Giới hạn:** N <= 30, Các số nguyên trong hộp có trị tuyệt đối <= $10^3$
**Dữ liệu vào:**

- Dòng đầu tiên nhập vào số nguyên dương N
- Tiếp theo là N nhóm dòng thể hiện lớp cắt của hình lập phương nhìn từ mặt trước, từ gần ra xa. Mỗi nhóm dòng gồm N dòng, theo thứ tự từ trên xuống dưới của hộp. Mỗi dòng gồm N số theo thứ tự trái sang phải của hộp.

**Kết quả:** In ra số nguyên duy nhất là tổng khối lập phương con lớn nhất tìm được.
**Ví dụ:**

| input | output |
|---|---|
| **3<br>0 -1 3 -5 7 4 -8 9 1 -1 -3 -1<br>2 -1 5 0 -1 3<br>3 1 -1<br>1 3 2<br>1 -2 1** | **27** |

## Bài 11. Biến đổi chuỗi.

Cho một xâu kí tự gồm các chữ cái latin, có in thường lẫn in hoa. Ta gọi chuỗi là đẹp nếu thỏa mãn một trong ba điều kiện sau đây:

- Toàn bộ kí tự của chuỗi đều là in thường
- Toàn bộ kí tự của chuỗi đều là in hoa
- Không có chữ in thường nào nằm trước chữ in hoa, nghĩa là tất cả chữ in hoa đều nằm đầu chuỗi. (ví dụ ABc là chuỗi đẹp, nhưng BaH thì không đẹp)

Mỗi thao tác bạn có thể thay đổi 1 kí tự bất kì trong chuỗi từ in thường thành in hoa hoặc in hoa thành in thường.
**Yêu cầu:** Hãy đếm số thao tác ít nhất để biến đổi chuỗi đã cho thành chuỗi đẹp.
**Giới hạn:** Chiều dài chuỗi nhập vào không quá $10^5$ kí tự.
**Dữ liệu vào:**

- Một chuỗi kí tự không rỗng duy nhất

**Kết quả:** In ra một số nguyên duy nhất là số thao tác ít nhất để biến chuỗi đã cho thành chuỗi đẹp.
**Ví dụ:**

| input | output |
|---|---|
| **GTmpijYHAaKA** | **5** |

## Bài 12. Xếp hàng mua vé

Có **N** người xếp hàng mua vé xem phim. Ta đánh số họ từ 1 đến N theo thứ tự từ đầu hàng đến cuối hàng. Mỗi người cần mua một là vé, người bán vé được phép bán cho mỗi người tối đa hai vé. Vì thế, một số người có thể rời hàng và nhờ người đứng trước mình mua hộ vé. Biết $T_i$ là thời gian cần thiết để người i mua xong vé cho mình. Nếu người i + 1 rời khỏi hàng và nhờ người i mua hộ vé thì thời gian để người thứ i mua cả hai vé là $R_i$.

**Yêu cầu:** Bạn hãy xác định xem những người nào cần rời hàng và nhờ người đứng trước mình mua hộ vé để tổng thời gian phục vụ bán vé là nhỏ nhất

**Giới hạn:** $N \le 10^5$ và $Ti, Ri \le 10^6$

**Dữ liệu vào:**

- Dòng đầu tiên, chứa một số nguyên dương N duy nhất
- Dòng thứ hai, gồm N số nguyên dương $T_1, T_2, \ldots T_n$
- Dòng cuối cùng, gồm N - 1 số nguyên dương $R_1, R_2, \ldots R_{n-1}$

**Kết quả:** In ra số nguyên duy nhất là tổng thời gian phục vụ bán vé nhỏ nhất.

**Ví dụ:**

| input | output |
|---|---|
| **5**<br>**2 6 9 10 4**<br>**4 9 7 10** | **15** |

## Bài 13. Dãy con tăng dài nhất

Dãy con của một dãy là dãy có thể đạt được bằng cách xóa đi một số phần tử trong dãy ban đầu, và giữ nguyên thứ tự những phần tử còn lại. Bài toán tìm dãy con tăng dài nhất là tìm một dãy con sao cho trong dãy con này, phần tử đứng sau lớn hơn hẵn phần tử đứng trước và số phần tử của dãy tìm được lớn nhất có thể.

**Yêu cầu:** Cho một dãy A có N phần tử. Tìm dãy con tăng dài nhất của dãy A.

**Giới hạn:** $N \le 10^3$ và ¿ $A_i \vee \le 10^9$

**Dữ liệu vào:**

- Dòng đầu tiên, chứa một số nguyên dương N duy nhất
- Dòng thứ hai, gồm N số nguyên dương $A_1, A_2, \ldots A_n$

**Kết quả:** In ra số nguyên duy nhất là độ dài dãy con tăng dài nhất của dãy đã cho.

**Ví dụ:**

| input | output |
|---|---|
| **5** | **3** |

| | |
|---|---|
| **5 1 3 2 6** | |

**Giải thích:** Dãy con tăng dài nhất là 1, 3, 6

## Bài 14. Trò chơi leo thang

Một cầu thang có N bậc thang được đánh số từ 1 tới N từ dưới lên trên. Bob có thể đi lên một bậc thang, hoặc nhảy một bước lên hai bậc thang. Tuy nhiên, có K bậc thang đã bị hỏng và Bob không thể đứng trên các bậc thang đó. Bob đang đứng ở bậc thang số 1 (bậc thang không bao giờ bị hỏng). Bờm đố bạn một câu hỏi: Có bao nhiêu cách để Bob leo đến bậc thang thứ N?

**Yêu cầu:** Đếm số cách Bob leo hết cầu thang. Vì kết quả rất lớn nên hãy in kết quả chia lấy dư cho $(10^9+7)$

**Giới hạn:** $K < N \le 10^5$

**Dữ liệu vào:**

- Dòng đầu tiên, chứa hai số nguyên dương N và K
- Dòng thứ hai, gồm K số nguyên dương là vị trí các bậc thang bị hỏng.

**Kết quả:** In ra số nguyên duy nhất là số cách Bob leo hết cầu thang chia lấy dư cho $(10^9 + 7)$ .

**Ví dụ:**

| input | output |
|---|---|
| **5 1**<br>**4** | **2** |

**Giải thích:**Cách 1: Bob đi từ 1 -> 2 -> 3 -> 5Cách 2: Bob đi từ 1 -> 3 -> 5

## Bài 15. Bài toán cái túi

Cho N đồ vật, đồ vật thứ i có trọng lượng là $W_i$ và có giá trị là $V_i$ . Một người được lấy miễn phí bất kì đồ vật nào (mỗi đồ vật chỉ được lấy tối đa 1 lần) trong N đồ vật này, người đó tất nhiên muốn lấy các đồ vật sao cho tổng giá trị mà anh ta thu được là lớn nhất, tuy nhiên thì anh ấy chỉ đem một cái túi có thể chứa được tối đa trọng lượng là M nên phải chọn các đồ vật sao cho tổng trọng lượng không được vượt quá M.

**Yêu cầu:** Hãy giúp anh ta chọn các đồ vật có tổng trọng lượng không quá M và đạt giá trị lớn nhất, in ra giá trị đó.

**Giới hạn:** N, M , $W_i$ , $V_i \le 10^3$

**Dữ liệu vào:**

- Dòng đầu tiên, chứa hai số nguyên dương N và M
- N dòng tiếp theo, mỗi dòng gồm hai số nguyên dương $W_i$ và $V_i$

**Kết quả:** In ra số nguyên duy nhất là kết quả bài toán.

**Ví dụ:**

| input | output |
|---|---|
| **5 10**<br>**3 20**<br>**1 19** | **63** |

| | |
|---|---|
| **7 30**<br>**3 24**<br>**6 15** | |

**Giải thích:** Chọn đồ vật 1, 2, 4.

Tổng thể tích là 3 + 1 + 3 và không quá 10

Tổng giá trị là 20 + 19 + 24 = 6

**Bài 16. Xâu con chung liên tiếp dài nhất**

Cho hai xâu s và t. Nói xâu t chứa s nếu tồn tại một xâu con các ký tự liên tiếp nhau của t bằng s. Ví dụ, với t = 'ABRACADABRA', nó sẽ chứa các xâu 'ABRA', 'RAC', 'D', 'ACADABRA', 'ABRACADABRA', nhưng không chứa xâu 'ABRC'. Xâu bất kỳ luôn luôn được coi là chứa xâu rỗng. Hai xâu X = 'ABRACADABRA' và Y = 'ECADADABRBCRDARA' cùng chứa các xâu 'CA', 'CADA', 'ADABR' và xâu rỗng. Đó là các xâu con chung cùng được X và Y chứa. Trong số các xâu con này, xâu 'ADABR' có độ dài lớn nhất (bằng 5).

**Yêu cầu:** Cho hai xâu X và Y chỉ chứa các ký tự la tinh in hoa. Hãy xác định độ dài lớn nhất của xâu con chung cùng được X và Y chứa.

**Giới hạn:** Độ dài mỗi xâu <= 10$_3$

**Dữ liệu vào:**

- Dòng đầu tiên, chứa xâu X
- Dòng thứ hai, chứa xâu Y

**Kết quả:** In ra số nguyên duy nhất là kết quả bài toán.

**Ví dụ:**

| input | output |
|---|---|
| ABRACADABRA<br>ECADADABRBCRDARA | 5 |

**Giải thích:** Xâu con chung liên tiếp dài nhất là **ADABR**

## Bài 17. Xâu con chung không liên tiếp dài nhất.

Cho xâu kí tự A có độ dài m, dãy xâu kí tự B có độ dài n. Tìm dãy con chung dài nhất của hai xâu này. Xâu con không liên tiếp hay còn gọi là dãy con.

Dãy con của một xâu là xâu thu được bằng cách xóa đi một số kí tự từ xâu ban đầu nhưng vẫn giữ nguyên thứ tự của các kí tự còn lại.

**Yêu cầu:** Cho hai xâu A và B chỉ chứa các ký tự la tinh in hoa. Hãy xác định độ dài lớn nhất của dãy con chung của A và B.

**Giới hạn:** m, n <= 10$_3$

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương m và n
- Dòng thứ hai, chứa xâu A có m kí tự

• Dòng thứ ba, chứa xâu B có n kí tự
**Kết quả:** In ra số nguyên duy nhất là kết quả bài toán.
**Ví dụ:**

| input | output |
|---|---|
| | 8 |

**Giải thích:** Xâu con chung không liên tiếp dài nhất là **DGBLRRXU**

## Bài 18. Dãy con tăng có tổng lớn nhất

Dãy con của một dãy là dãy có thể đạt được bằng cách xóa đi một số phần tử trong dãy ban đầu, và giữ nguyên thứ tự những phần tử còn lại. Cho dãy A có N phần tử nguyên dương $A_1, A_2, ..., A_n$.

**Yêu cầu:** Hãy tìm dãy con tăng có tổng các phần tử của dãy con đó là lớn nhất.
**Giới hạn:** N <= 5000 và $A_i <= 10^9$
**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- Dòng tiếp theo chứa N số nguyên dương $A_i$

**Kết quả:** In ra số nguyên duy nhất là tổng dãy con tăng mà bạn tìm được.
**Ví dụ:**

| input | output |
|---|---|
| 6<br>4 1 2 5 3 7 | 16 |

**Giải thích:** Dãy con cần tìm là 4, 5, 7

## Bài 19. Dãy con thân thiện

Dãy con của một dãy là dãy có thể đạt được bằng cách xóa đi một số phần tử trong dãy ban đầu, và giữ nguyên thứ tự những phần tử còn lại. Dãy thân thiện là dãy tăng (số sau lớn hơn số trước) mà hai phần tử liên tiếp là hai số nguyên tố cùng nhau. Hai số nguyên tố cùng nhau là hai số mà ước chung lớn nhất của chúng bằng 1.

Cho dãy A có N phần tử nguyên dương $A_1, A_2, ..., A_n$.
**Yêu cầu:** Hãy tìm dãy con có độ dài lớn nhất là dãy thân thiện.
**Giới hạn:** N <= 5000 và $A_i <= 10^9$
**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- Dòng tiếp theo chứa N số nguyên dương $A_i$

**Kết quả:** In ra số nguyên duy nhất là dãy con thân thiện mà bạn tìm được.
**Ví dụ:**

| input | output |
|---|---|

| | |
|---|---|
| **5**<br>**2 3 4 6 9** | **4** |

**Giải thích:** Dãy con thân thiện dài nhất là 2, 3, 4, 9

## Bài 20. Chơi thể thao

Hải là một vận động viên thể thao. Mỗi năm, có N giải đấu đánh số từ 1 đến N. Giải đấu thứ i được tổ chức vào ngày $A_i$ và nếu Hải tham gia sẽ được nhận $B_i$ tiền thưởng. Tuy nhiên, để đảm bảo sức khỏe cho Hải, huấn luyện viên quyết định hai giải đấu mà Hải tham gia phải cách nhau ít nhất K ngày ( $|A_i - A_j| >= K$).

**Yêu cầu:** Bạn hãy giúp Hải chọn giải đấu nào cần tham gia sao cho tổng tiền thưởng là lớn nhất có thể.

**Giới hạn:** $N \le 200, K \le 20, A_i \le 365, B_i \le 10^9$

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương N và K
- Dòng thứ hai, chứa dãy tăng dần N số nguyên dương $A_i$
- Dòng thứ ba, chứa dãy N số nguyên dương $B_i$

**Kết quả:** In ra số nguyên duy nhất là số tiền lớn nhất mà Hải có được.

**Ví dụ:**

| input | output |
|---|---|
| **5 2**<br>**1 2 3 4 5**<br>**1 5 1 5 1** | **10** |

## Bài 21. Trò chơi điện tử.

Cho 1 dãy số gồm N phần tử, mỗi phần tử có 1 giá trị nằm trong khoảng [$-10^9, 10^9$]. Ban đầu, bạn sẽ ở vị trí ô số 0 với tổng điểm là 0. Mỗi nước đi, người chơi có thể di chuyển sang phải tối thiểu là 1 bước và tối đa là K bước. Khi dừng lại ở 1 ô nào đó thì giá trị của ô đó sẽ được cộng vào tổng điểm. Bạn có thể dừng cuộc chơi bất cứ lúc nào. Hãy tìm cách chơi sao cho tổng điểm nhận được là nhiều nhất.

**Yêu cầu:** Nhập vào dãy A có N phần tử, in ra tổng điểm lớn nhất theo nguyên tắc trò chơi.

**Giới hạn:** $N <= 10^5$, $K < 100$

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương N và K
- Dòng thứ hai, chứa xâu N số nguyên $A_i$

**Kết quả:** In ra số nguyên duy nhất là số điểm lớn nhất có thể đạt được.

**Ví dụ:**

| input | output |
|---|---|
| **5 2** | **4** |

| | |
|---|---|
| -2 3 -6 -4 5 | |

## Bài 22. Sắp xếp bò đực cái.

Anh nông dân Bob có một đàn bò gồm rất nhiều con cái và con đực. Trong một hội chợ, anh muốn sắp một hàng bò gồm n con. Tuy nhiên những con bò đực rất hung hăng nếu đứng gần nhau, anh phải sắp tối thiểu k con bò cái xen giữa hai con bò đực để chúng khỏi húc nhau.

Bạn hãy giúp anh Bo đếm thử xem có bao nhiêu cách để sắp một hàng gồm n con bò mà hai con bò đực bất kỳ không húc nhau (anh Bob có rất nhiều bò nên không sợ thiếu bò cái hoặc bò đực). Ví dụ với n = 4 và k = 1, ta có 8 cách xếp như sau (M: bò đực, F: bò cái): FFFF, MFFF, FMFF, FFMF, FFFM, MFMF, MFFM, FMFM.

**Yêu cầu:** Đếm bao nhiêu cách sắp xếp bò để có tối thiểu k bò cái xen giữa 2 con bò đực.

**Giới hạn:** N, K <= $10^4$

**Dữ liệu vào:**

- Dòng duy nhất chứa hai số nguyên dương N và K

**Kết quả:** In ra số nguyên duy nhất là số cách sắp xếp bò. Vì kết quả rất lớn nên hãy chia lấy dư cho ($10^9 + 7$)

**Ví dụ:**

| input | output |
|---|---|
| 4 1 | 8 |

## Bài 23. Không bốc hai liên tiếp.

Cô giáo xếp N món quà lên bàn. Món quà thứ i có giá trị là $A_i$. Hải được phép bốc một số món quà trên bàn. Điều kiện là không được bốc hai món quà liên tiếp. **Yêu cầu:** Bạn hãy giúp Hải tính xem nên bốc như thế nào để đạt được nhiều điểm nhất.

**Giới hạn:** N <= $10^5$ và $A_i$ <= $10^5$

**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- Dòng tiếp theo, chứa N số nguyên dương $A_1, A_2, \ldots, A_n$

**Kết quả:** In ra số nguyên duy nhất là số điểm tối đa mà Hải có được.

**Ví dụ:**

| input | output |
|---|---|
| 5<br>5 7 1 1 6 | 13 |
| 7<br>3 1 9 8 5 1 7 | 24 |

**Giải thích:**

- Ví dụ 1, bốc món quà vị trí 2, 5

- Ví dụ 2, bốc món quà vị trí 1, 3, 5, 7

## Bài 24. Trò chơi của Bờm.

Bờm rất thì chơi trò chơi. Một hôm, cậu bé nghĩ ra một trò chơi rất thú vị và quyết định chơi nó.

Cho một dãy A gồm N số nguyên. Người chơi có thể thực hiện một số bước. Trong một bước duy nhất có thể chọn một phần tử của dãy là $A_k$ và xóa nó, sau đó tất cả phần tử có giá trị bằng với $A_k + 1$ và $A_k - 1$ cũng phải bị xóa khỏi chuỗi. Sau khi xóa vẫn giữ nguyên vị trí những phần tử khác. Bước đó mang lại $A_k$ điểm cho người chơi. Bờm rất tham lam, vì vậy cậu bé muốn đạt được nhiều điểm nhất có thể.

**Yêu cầu:** Bạn hãy giúp Bờm tính xem trong trường hợp chơi tối ưu nhất thì Bờm có bao nhiêu điểm.

**Giới hạn:** N <= $10^5$ và $A_i$ <= $10^5$

**Dữ liệu vào:**

• Dòng đầu tiên chứa số nguyên dương N

• Dòng tiếp theo, chứa N số nguyên dương $A_1, A_2, \dots, A_n$ **Kết quả:** In ra số nguyên duy nhất là số điểm tối đa mà Bờm có được.

**Ví dụ:**

| input | output |
|---|---|
| **5**<br>**3 1 3 2 4** | **7** |

**Giải thích:**

- Chọn k=3, được3 điểm, xóa $A_4$, $A_5$
- Chọn k=1, được3 điểm, không có phần tử nào đểxóa
- Chọn k=2, được1 điểm, không có phần tử nào đểxóa

## Bài 25. Không bốc ba liên tiếp.

Cô giáo xếp N món quà lên bàn. Món quà thứ i có giá trị là $A_i$. Hải được phép bốc một số món quà trên bàn. Điều kiện là không được bốc ba món quà liên tiếp. **Yêu cầu:** Bạn hãy giúp Hải tính xem nên bốc như thế nào để đạt được nhiều điểm nhất.

**Giới hạn:** N <= $10^5$ và $A_i$ <= $10^5$

**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- Dòng tiếp theo, chứa N số nguyên dương $A_1, A_2, \dots, A_n$

**Kết quả:** In ra số nguyên duy nhất là số điểm tối đa mà Hải có được.

**Ví dụ:**

| input | output |
|---|---|
| **4**<br>**9 3 5 4** | **18** |

**Giải thích:** Bốc món quà vị trí 1, 3, 4

## Bài 26. Không bốc K liên tiếp.

Cô giáo xếp N món quà lên bàn. Món quà thứ i có giá trị là $A_i$. Hải được phép bốc một số món quà trên bàn. Điều kiện là không được bốc K món quà liên tiếp. **Yêu cầu:** Bạn hãy giúp Hải tính xem nên bốc như thế nào để đạt được nhiều điểm nhất.

**Giới hạn:** $N <= 10^5$ và $A_i <= 10^5$ và $K <= 100$

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương N và K
- Dòng tiếp theo, chứa N số nguyên dương $A_1, A_2, \ldots, A_n$

**Kết quả:** In ra số nguyên duy nhất là số điểm tối đa mà Hải có được.

**Ví dụ:**

| input | output |
|---|---|
| **9 5**<br>**9 3 5 4 2 5 3 6 8** | **43** |

**Giải thích:** Bốc món quà vị trí 1, 2, 3, 4, 6, 7, 8, 9

## Bài 27. Đếm số đường đi

Cho mê cung có N x N ô vuông, mỗi ô có thể có bẫy, không thể đứng trên ô có bẫy. Nếu là ô trống thì bạn có thể đứng. Ô có bẫy kí hiệu là #. Ô trống kí hiệu là **.** Bạn đang ở ô (1, 1) là ô ở góc trái trên và không bao giờ có bẫy. Bạn cần đến ô (N, N) ở góc phải dưới để kết thúc trò chơi. Bạn có thể đi sang phải hoặc xuống dưới. **Yêu cầu:** Hãy đếm số đường đi thỏa mãn không được đạp bẫy và kết thúc tại ô (N, N). Vì kết quả rất lớn nên hãy chia lấy dư cho ($10^9 + 7$)

**Giới hạn:** $N <= 10^3$

**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- N dòng tiếp theo, mỗi dòng chứa N kí tự:
  - # là ô có bẫy
  - **.** là ô không có bẫy

**Kết quả:** In ra số nguyên duy nhất là số điểm tối đa mà Hải có được.

**Ví dụ:**

| input | output |
|---|---|
| **4**<br>  *****<br>    *****<br>***** | **3** |

**Giải thích:** Cách 1: (1,1)-> (2, 1) -> (3, 1) -> (3, 2) -> (3, 3) -> (4,3) -> (4, 4)

Cách 2: (1, 1) -> (2, 1)->(3, 1) -> (3, 2) -> (4, 2) -> (4, 3) ->(4, 4)
Cách 3: (1, 1) -> (1, 2)->(1, 3) -> (2, 3) -> (3, 3) -> (4, 3) ->(4, 4)

## Bài 28. Đường đi tối ưu 1

Cho một bàn cờ hình chữ nhật gồm m hàng và n cột. Mỗi ô trên bàn cờ này có ghi một giá trị nguyên. Xuất phát từ ô (1, 1) ở góc trái trên, bạn cần di chuyển đến ô (m, n) ở góc phải dưới. Ở mỗi bước, bạn được di chuyển sang phải một ô hoặc xuống dưới một ô. Hãy tìm cách di chuyển để tổng giá trị của các ô trên đường đi là lớn nhất.

**Yêu cầu:** Hãy in ra giá trị đường đi có tổng lớn nhất.

**Giới hạn:** m, n <= $10_3$ và $|A_{ij}|$ <= $10_9$

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương m và n
- m dòng tiếp theo, mỗi dòng chứa n số nguyên $A_{ij}$.

**Kết quả:** In ra số nguyên duy nhất là số điểm tối đa mà Hải có được.

**Ví dụ:**

| input | output |
|---|---|
| **3 3**<br>**1 5 -3**<br>**-2 -4 -6**<br>**10 -5 1** | **5** |

## Bài 29. Phân trang

Văn bản là một dãy gồm N từ đánh số từ 1 đến N. Từ thứ i có độ dài là $W_i$. Phân trang là một cách xếp lần lượt các từ của văn bản vào dãy các dòng, mỗi dòng có độ dài tối đa là L: sao cho tổng độ dài của các từ trên cùng một dòng không vượt quá L.

Ta gọi hệ số phạt của mỗi dòng trong cách phân trang là hiệu số **L - S**, trong đó **S** là tổng độ dài của các từ xếp trên dòng đó. Hệ số phạt của cách phân trang là giá trị lớn nhất trong số các hệ số phạt của các dòng.

**Yêu cầu:** Bạn hãy lập chương trình nhận đầu vào là danh sách các từ, tìm cách phân trang với hệ số phạt nhỏ nhất.

**Giới hạn:** N, L <= $10^3$ và $W_i$ <= L

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương N và L
- N dòng tiếp theo, dòng thứ i là số nguyên dương $W_i$

**Kết quả:** In ra số nguyên duy nhất là số điểm tối đa mà Hải có được.

**Ví dụ:**

| input | output |
|---|---|
| **4 5**<br>**3**<br>**2** | **2** |

| | |
|---|---|
| **2**<br>**4** | |

**Giải thích:** Trong ví dụ trên ta có thể phân thành 3 dòng:

- Dòng 1: gồm một từ (3) hệ số phạt dòng này là 5 - 3 = 2
- Dòng 2: gồm hai từ (2, 2) hệ số phạt là 5 - 4 = 1
- Dòng 3: gồm một từ (4) hệ số phạt là 5 - 4 = 1
- Đáp án: max(2, 1, 1) = 2

## Bài 30. Dãy con dài nhất có tổng chia hết cho K

Cho một dãy gồm N số nguyên dương $A_1$, $A_2$,  $A_n$ và số nguyên dương K. Dãy con của một dãy là dãy có thể đạt được bằng cách xóa đi một số phần tử trong dãy ban đầu, và giữ nguyên thứ tự những phần tử còn lại

**Yêu cầu:** Hãy tìm dãy con gồm nhiều phần tử nhất của dãy đã cho sao cho tổng các phần tử của dãy con này chia hết cho K.

**Giới hạn:** K <= 100 và $A_i$ <= $10^9$

- Subtask 1: N <= $10^3$
- Subtask 2: N <= $10^5$

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương N và K
- Dòng tiếp theo chứa N số nguyên dương $A_i$

**Kết quả:** In ra số nguyên duy nhất là độ dài dãy con dài nhất có tổng chia hết cho K.

**Ví dụ:**

| input | output |
|---|---|
| **10 3**<br>**2 3 5 7 9 6 12 7 11 15** | **9** |

**Giải thích:** Có nhiều dãy con dài nhất có tổng chia hết cho 3.
Một trong số đó là dãy 2, 3, 5, 7, 9, 6, 12, 7, 15

## Bài 31. Đếm số biểu thức có tổng bằng S

Cho một dãy gồm N số nguyên dương $A_1$, $A_2$, $A_n$ có giá trị phân biệt. Và một số S. Một biểu thức trong bài này chỉ bao gồm phép cộng, hai biểu thức khác nhau khi khác số số hạng hoặc có 1 vị trí mà số hạng ở biểu thức này khác số hạng ở biểu thức kia.

**Yêu cầu:** Hãy đếm số biểu thức được tạo thành từ những số trong dãy A, và có tổng bằng S.

**Giới hạn:** N <= 100, S <= $10^5$ và $A_i$ <= $10^5$

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương N và S
- Dòng tiếp theo chứa N số nguyên dương $A_i$

**Kết quả:** In ra số nguyên duy nhất là độ dài dãy con dài nhất có tổng chia hết cho K.

**Ví dụ:**

| input | output |
|---|---|
| 3 9<br>2 3 5 | 8 |

**Giải thích:**

- 2 + 2 + 5
- 2 + 5 + 2
- 5 + 2 + 2
- 3 + 3 + 3
- 2 + 2 + 2 + 3
- 2 + 2 + 3 + 2
- 2 + 3 + 2 + 2
- 3 + 2 + 2 + 2

## Bài 32. Dãy bị khuyết

Cho một dãy gồm N số nguyên dương $A_1$, $A_2$, $A_n$. Nhưng đã bị khuyết một số vị trí, ở đấy, $A_i = 0$. Một dãy đẹp là dãy mà 2 phần tử cạnh nhau, chênh lệch nhiều nhất 1 đơn vị.

**Yêu cầu:** Hãy đếm số cách điền các giá trị trong phạm vi [1, M] vào ô bị khuyết, sao cho dãy mới không có ô khuyết và là một dãy đẹp.

**Giới hạn:** $N <= 10^5$, $M <= 100$ và $A_i <= M$

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương N và M
- Dòng tiếp theo chứa N số nguyên dương $A_i$ . Nếu $A_i = 0$, có nghĩa là vị trí i bị khuyết.

**Kết quả:** In ra số nguyên duy nhất là kết quả bài toán lấy dư cho ($10_9 + 7$).

**Ví dụ:**

| input | output |
|---|---|
| 3 5<br>2 0 2 | 3 |

**Giải thích:** Các cách điền tạo thành dãy mới là {2, 1, 2} , {2, 2, 2}, {2, 3, 2}

## Bài 33. Thuê nhà

Hải có một ngôi nhà sang trọng, tiện nghi đầy đủ. Sau khi đăng tin cho thuê nhà, Hải nhận được N đơn thuê nhà, đơn thứ i thuê nhà từ sáng ngày $A_i$ đến tối ngày $B_i$. Vì đang muốn tăng thêm uy tín, Hải chỉ quan tâm đến số đơn thuê nhà mà Hải có thể đáp ứng.

**Yêu cầu:** Hãy giúp Hải nên nhận đơn thuê nhà nào, những người ở không đụng ngày nhau và số đơn cho thuê là nhiều nhất.

**Giới hạn:** $N <= 10^3$, $1 <= A_i <= B_i <= 10^9$
**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- N dòng tiếp theo, dòng thứ i chứa hai số nguyên dương $A_i$ và $B_i$.

**Kết quả:** In ra số nguyên duy nhất là số đơn thuê nhiều nhất mà Hải có thể đáp ứng.
**Ví dụ:**

| input | output |
|---|---|
| 5<br>3 5<br>1 2<br>8 9<br>2 3<br>4 6 | 3 |

**Giải thích:** Hải đáp ứng đơn [3, 5] ; [1,2] ; [8,9]

## Bài 34. Dãy con dài nhất hình chữ V

Dãy con của một dãy là dãy có thể đạt được bằng cách xóa đi một số phần tử trong dãy ban đầu, và giữ nguyên thứ tự những phần tử còn lại. Dãy hình chữ V là dãy mà các phần tử đầu là một dãy giảm dần đến 1 phần tử ở đáy là nhỏ nhất, sau đó tăng dần. Ví dụ dãy số 5 3 2 1 4 6 là dãy hình chữ V. Cho một dãy A có N phần tử nguyên dương.
**Yêu cầu:** Hãy viết chương trình tìm dãy con dài nhất hình chữ V của dãy đã cho.
**Giới hạn:** $N <= 10^3$, $1 <= A_i <= 10^9$
**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- Dòng thứ hai, chứa N số nguyên dương $A_i$.

**Kết quả:** In ra số nguyên duy nhất là độ dài tối đa của dãy con hình chữ V của dãy đã cho.
**Ví dụ:**

| input | output |
|---|---|
| 6<br>5 3 1 1 2 6 | 5 |

**Giải thích:** Dãy con dài nhất hình chữ V là 5, 3, 1, 2, 6

Trò chơi với băng số là trò chơi tham gia trúng thưởng được mô tả như sau: Có một băng hình chữ nhật được chia ra làm N ô vuông, đánh số từ trái qua phải bắt đầu từ 1. Trên ô vuông thứ i người ta ghi một số nguyên dương $A_i$ với i = 1, 2, N. Ở một lượt chơi, người tham gia trò chơi được quyền lựa chọn một số lượng tùy ý các ô trên băng số. Giả sử theo thứ tự từ trái qua phải, người chơi lựa chọn các ô $i_1$, $i_2$, ..., $i_k$. Khi đó điểm số mà người chơi đạt được sẽ là: $a_{i1}$ - $a_{i2}$ + ... + $(-1)^{k-1}$ $a_{ik}$.

**Yêu cầu:** Hãy tính số điểm lớn nhất có thể đạt được từ một lượt chơi.

**Giới hạn:** N <= $10_6$, 1 <= $A_i$ <= $10_4$

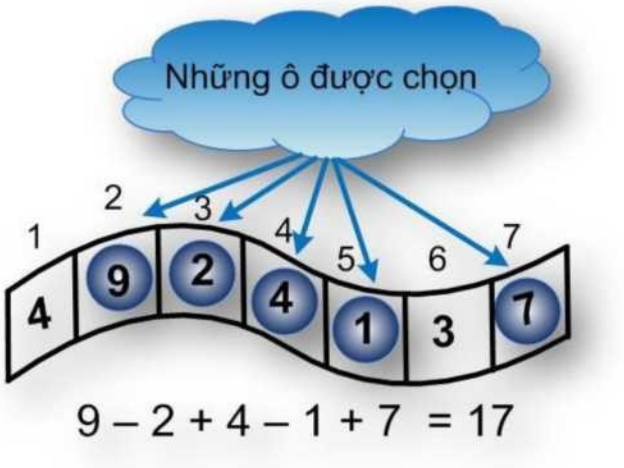
**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- Dòng thứ hai, chứa N số nguyên dương $A_i$.

**Kết quả:** In ra số nguyên duy nhất kết quả của bài toán.

**Ví dụ:**

**Giải thích:**




| input | output |
|---|---|
| **7**<br>**4 9 2 4 1 3 7** | **17** |

## Bài 36. Nối mạng

Các học sinh khi đến thực tập trong phòng máy tính thường hay chơi trò chơi điện tử trên mạng. Để ngăn ngừa, người trực phòng máy đã ngắt tất cả các máy tính ra khỏi mạng và xếp chúng thành một dãy trên một cái bàn dài và gắn chặt máy xuống mặt bàn rồi đánh số thứ tự các máy từ 1 đến N theo chiều từ trái sang phải. Các học sinh tinh nghịch không chịu thua, họ đã quyết định tìm cách nối các máy trên bàn bởi các đoạn dây nối sao cho mỗi máy được nối với ít nhất một máy khác. Để tiến hành công việc này, họ đã đo khoảng cách giữa hai máy liên tiếp.

**Yêu cầu:** Bạn hãy giúp các học sinh này tìm cách nối mạng thoả mãn yêu cầu đặt ra sao cho tổng độ dài cáp nối phải sử dụng là ít nhất.

**Giới hạn:** N <= $10_5$, 1 <= $A_i$ <= $10_4$

**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- Dòng thứ hai, chứa N - 1 số nguyên dương $A_i$ là khoảng cách máy i đến i+1

**Kết quả:** In ra số nguyên duy nhất là độ dài của cáp nối cần sử dụng.

**Ví dụ:**

| input | output |
|---|---|
| **6**<br>**2 2 3 2 2** | **7** |

**Giải thích :** Nối máy 1 với máy 2 (tốn 2), nối máy 3 với máy 4 (tốn 3), nối máy 5 với máy 6 (tốn 2). Tổng độ dài là 7.

## Bài 37. Cắt bìa

Cho một miếng bìa hình chữ nhật kích thước M x N là hai số nguyên, hãy cắt hình chữ nhật đó thành các hình vuông. Mỗi lần cắt, bạn chọn một hình chữ nhật và chia chúng thành hai hình chữ nhật sao cho độ dài cạnh vẫn là các số nguyên.

**Yêu cầu:** Hỏi số lượt cắt nhỏ nhất là bao nhiêu để có thể được các mảnh bìa hình vuông?

**Giới hạn:** M, N <= 500

**Dữ liệu vào:**

• Dòng duy nhất chứa hai số nguyên dương M và N **Kết quả:** In ra số nguyên duy nhất là số lượt cắt ít nhất.

**Ví dụ:**

| input | output |
|---|---|
| **3 5** | **3** |
| **5 6** | **4** |

**Giải thích :**

## Bài 38. Nhảy cóc 1

Có N hòn đá, được đánh số từ 1 đến N. Với mỗi chỉ số i (1 <= i <= N), độ cao của hòn đá thứ i là $H_i$

Ban đầu, có một chú ếch đang ngồi ở hòn đá thứ nhất và chú sẽ thực hiện liên tục một loạt các hành động sau: Nếu chú đang ngồi ở hòn đá thứ i chú có thể nhảy đến hòn đá thứ i + 1 hoặc hòn đá thứ i + 2. Chú sẽ mất chi phí nhảy là $|H_i - H_j|$ với j là hòn đá mà chú ếch đáp xuống.

**Yêu cầu:** Hỏi chi phí tối thiểu để chú ếch nhảy từ hòn đá thứ nhất đến hòn đá thứ N là bao nhiêu?

**Giới hạn:** $N <= 10^5$ và $H_i <= 10^4$

**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên N
- Dòng thứ hai, chứa N số nguyên dương $H_i$ là độ cao hòn đá thứ i

**Kết quả:** In ra số nguyên duy nhất là chi phí ít nhất để chú ếch nhảy đến hòn đá N. **Ví dụ:**

| input | output |
|---|---|
| **4**<br>**10 30 40 20** | **30** |

**Giải thích:** Một đường đi tối ưu là 1 -> 2 -> 4

Chi phí là |10 - 30| + |30 - 20| = 30

## Bài 39. Nhảy cóc 2

Có N hòn đá, được đánh số từ 1 đến N. Với mỗi chỉ số i (1 <= i <= N), độ cao của hòn đá thứ i là $H_i$

Ban đầu, có một chú ếch đang ngồi ở hòn đá thứ nhất và chú sẽ thực hiện liên tục một loạt các hành động sau: Nếu chú đang ngồi ở hòn đá thứ i chú có thể nhảy đến hòn đá thứ i + 1, i + 2, i + 3, ..., i + K. Chú sẽ mất chi phí nhảy là $|H_i - H_j|$ với j là hòn đá mà chú ếch đáp xuống.

**Yêu cầu:** Hỏi chi phí tối thiểu để chú ếch nhảy từ hòn đá thứ nhất đến hòn đá thứ N là bao nhiêu?

**Giới hạn:** $N <= 10^5$, $K <= 100$ và $H_i <= 10^4$

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương N và K
- Dòng thứ hai, chứa N số nguyên dương $H_i$ là độ cao hòn đá thứ i

**Kết quả:** In ra số nguyên duy nhất là chi phí ít nhất để chú ếch nhảy đến hòn đá N.

**Ví dụ:**

| input | output |
|---|---|
| **5 3**<br>**10 30 40 50 20** | **30** |

**Giải thích:** Một đường đi tối ưu là 1 -> 2 -> 5

Chi phí là |10 - 30| + |30 - 20| = 30

## Bài 40. Vận động viên ba môn.

Hải là một vận động viên chuyên nghiệp, chơi được cả ba môn: Bóng đá, Cầu lông, Điền kinh. Một giải Olympic thể thao diễn ra trong N ngày liên tiếp, Hải đăng kí thi 3 môn này. Mỗi ngày Hải chỉ tham gia được đúng 1 môn thể thao, nếu ngày thứ i,

- Hải chơi Bóng đá, sẽ nhận được $B_i$ tiền
- Hải chơi Cầu lông, sẽ nhận được $C_i$ tiền
- Hải chạy Điền kinh, sẽ nhận được $D_i$ tiền

Vì lí do sức khỏe, Hải không thể tham gia 2 môn thể thao giống nhau vào hai ngày liên tiếp

**Yêu cầu:** Hỏi số tiền tối đa Hải có thể nhận được là bao nhiêu?

**Giới hạn:** $N <= 10^5$, $B_i$, $C_i$, $D_i <= 10^4$

**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- N dòng tiếp theo, mỗi dòng gồm ba số nguyên dương $B_i$ , $C_i$ , $D_i$ **Kết quả:**

In ra số nguyên duy nhất là số tiền tối đa Hải nhận được.

**Ví dụ:**

| input | output |
|---|---|
| **3**<br>**10 40 70**<br>**20 50 80**<br>**30 60 90** | **210** |

**Giải thích:** Hải tham gia theo thứ tự: Điền kinh, Cầu lông, Điền kinh. Và nhận 70 + 50 + 90 tiền

## Bài 41. Chia kẹo cho hai bé.

Bờm và Tý là đôi bạn thân. Một hôm, cô giáo cho N gói kẹo, gói thứ i có $A_i$ viên kẹo. Cô muốn cho Bờm và Tý tất cả N gói kẹo này, không được xé gói kẹo (Để hai bạn mang về), và phải cho hết, không được giữ lại. Cô muốn chia N gói kẹo thành hai tập cho Bờm và Tý. Bờm và Tý là đôi bạn thân, nên nếu số viên kẹo nhận được chênh lệch quá lớn, hai bạn sẽ xảy ra mâu thuẫn, mất tình bạn.

**Yêu cầu:** Hãy chia N gói kẹo thành hai tập cho Bờm và Tý sao cho chênh lệch số viên kẹo của Bờm và Tý là nhỏ nhất.

**Giới hạn:** N <= $10_2$ , $A_i$ <= $10_2$

**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- Dòng tiếp theo chứa N số nguyên dương $A_i$

**Kết quả:** In ra số nguyên duy nhất là chênh lệch tối thiếu của Bờm và Tý.

**Ví dụ:**

| input | output |
|---|---|
| **5**<br>**1 5 2 3 6** | **1** |

**Giải thích:** Số kẹo của Bờm là 1 + 5 + 3 và Số kẹo của Tý là 2 + 6

## Bài 42. Đổi tiền

Đất nước Byteland có N loại tiền. Loại thứ i có mệnh giá là $A_i$ byte, số lượng rất lớn, đủ để đổi cho khách hàng. Một khách du lịch đến đem 1 thỏi vàng có mệnh giá bằng M byte. Du khách muốn đổi sang tiền byteland để dễ dàng sử dụng.

**Yêu cầu:** Hãy giúp du khách đổi thỏi vàng giá trị M byte sang tiền byteland sao cho số tờ tiền là ít nhất, để dễ dàng mang theo.

**Giới hạn:** N <= 100 , M <= $10_4$ và $A_i$ <= $10_4$

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương N và M
- Dòng tiếp theo chứa N số nguyên dương $A_i$

**Kết quả:** Nếu không thể đổi được, hãy in ra -1

Ngược lại, in ra số tờ tiền ít nhất để đổi được thỏi vàng giá trị M byte.

**Ví dụ:**

| input | output |
|---|---|
| **3 9**<br>**1 2 3** | **3** |

**Giải thích:** Du khách đổi 3 tờ giá trị 3 byte.

## Bài 43. Dãy con chung dài nhất

Dãy con của một dãy là dãy có thể đạt được bằng cách xóa đi một số phần tử trong dãy ban đầu, và giữ nguyên thứ tự những phần tử còn lại Cho M dãy, mỗi dãy là một hoán vị của N số tự nhiên đầu tiên.

**Yêu cầu:** Hãy tìm độ dài dãy con chung dài nhất của M dãy đó.

**Giới hạn:** N <= 1000 và M <= 10

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương N và M
- M dòng tiếp theo, mỗi dòng chứa N số tự nhiên là dãy hoán vị của N số tự nhiên đầu tiên.

**Kết quả:** Nếu không thể đổi được, hãy in ra -1

Ngược lại, in ra số tờ tiền ít nhất để đổi được thỏi vàng giá trị M byte.

**Ví dụ:**

| input | output |
|---|---|
| **5 3**<br>**1 5 3 4 2**<br>**1 3 4 2 5**<br>**3 1 5 4 2** | **3** |

**Giải thích:**

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

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

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

## Bài 44. Hình vuông cùng màu lớn nhất

Cho bảng M x N chia thành lưới các ô vuông đơn vị. Có M hàng và N cột. Trên các ô vuông đơn vị được tô màu trắng hoặc đen.

**Yêu cầu:** Hãy tìm hình vuông lớn nhất gồm các ô vuông nhỏ có chung màu.

**Giới hạn:** M, N <= 1000

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương M và N
- M dòng tiếp theo, mỗi dòng chứa N số:
    - Ô (i, j) màu trắng nếu $A_{ij} = 0$
    - Ô (i, j) màu đen nếu $A_{ij} = 1$

**Kết quả:** In ra kích thước cạnh lớn nhất của hình vuông màu đấy.

**Ví dụ:**

| input | output |
|---|---|
| 11 13<br>0 0 0 0 0 1 0 0 0 0 0 0 0<br>0 0 0 0 1 1 1 0 0 0 0 0 0<br>0 0 1 1 1 1 1 1 1 0 0 0 0<br>0 0 1 1 1 1 1 1 1 0 0 0 0<br>0 1 1 1 1 1 1 1 1 1 0 0 0<br>1 1 1 1 1 1 1 1 1 1 1 0 0<br>0 1 1 1 1 1 1 1 1 1 0 0 0<br>0 0 1 1 1 1 1 1 1 0 0 0 0<br>0 0 1 1 1 1 1 1 1 0 0 0 0<br>0 0 0 0 1 1 1 0 0 0 0 1 1<br>0 0 0 0 0 1 0 0 0 0 0 1 1 | 7 |

## Bài 45. Thuê xe du lịch

Hãng xe du lịch XYZ có N loại xe có chất lượng và giá thành khác nhau (Số lượng xe không hạn chế), loại xe thứ i có thể chở được $A_i$ khách và giá thành là $B_i$ cho mỗi chuyến đi. Một đoàn khách du lịch gồm M người cần thuê xe để đi du lịch.

**Yêu cầu:** Hãy tìm cách thuê xe để tổng chi phí phải trả là ít nhất.

**Giới hạn:** N , $A_i$ , $B_i$ <= 200 và M <= 40000

**Dữ liệu vào:**

- Dòng đầu tiên chứa hai số nguyên dương N và M
- N dòng tiếp theo, mỗi dòng chứa hai số nguyên dương $A_i$ và $B_i$

**Kết quả:** In ra chi phí nhỏ nhất để thuê xe chở được M người.

**Ví dụ:**

| input | output |
|---|---|
| 5 100<br>40 40<br>30 7<br>40 6<br>23 48 | 18 |

| | |
|---|---|
| **12 4** | |

**Giải thích:** Thuê 3 xe loại 3 có thể chở được 120 người (đủ chở 100 người) và tổng chi phí là 18

## Bài 46. Xây tháp ngoài hành tinh

TA cần xây một tòa tháp ngoài hành tinh. TA có N loại khối đá, loại thứ i có chiều cao là $H_i$, có số lượng là $C_i$ và sức chịu là $A_i$ (Vì ở hành tinh này, càng lên cao càng nóng, nếu khối đá có sức chịu X, mà một phần nào của khối đá vượt quá độ cao X sẽ bị hỏng cả tòa tháp).

**Yêu cầu:** Hãy giúp TA sắp xếp khối nào dưới, khối nào trên để độ cao của tòa tháp là cao nhất có thể.

**Giới hạn:** N <= 400, $H_i$ <= 100 , $C_i$ <= 10 , $A_i$ <= 40000

**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- N dòng tiếp theo, mỗi dòng chứa 3 số nguyên dương $H_i$ , $A_i$ , $C_i$ **Kết quả:**

In ra chi phí nhỏ nhất để thuê xe chở được M người.

**Ví dụ:**

| input | output |
|---|---|
| **3**<br>**7 40 3**<br>**5 23 8**<br>**2 52 6** | **48** |

**Giải thích:** Ta xếp từ dưới lên trên:

- 3 khốiloại2 giúp tòa tháp lên15
- 3 khốiloại1 giúp tòa tháp lên36
- 6 khốiloại3 giúp tòa tháp lên48

## Bài 47. Bốc sỏi

Cô giáo cho Hải và Hùng chơi 1 trò chơi. Cô rải N viên sỏi thẳng hàng, viên sỏi thứ i ghi số nguyên $A_i$ . Mỗi lượt chơi, một người sẽ bốc một viên sỏi đầu hàng hoặc cuối hàng và nhận được điểm bằng số nguyên ghi trên sỏi, nếu đó là số nguyên âm, người chơi sẽ bị trừ điểm, và điểm có thể âm. Hai người đều nhìn thấy số ghi trên đó và chơi sao cho điểm lớn nhất có thể. Hải được chơi lượt đầu.

**Yêu cầu:** Hãy tính xem Hải được bao nhiêu điểm.

**Giới hạn:** N <= 5000 và | $A_i$ | <= $10^9$

**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- Dòng tiếp theo chứa N số nguyên $A_i$

**Kết quả:** In ra số điểm của Hải khi cả hai người đều chơi tối ưu.

**Ví dụ:**

| input | output |
|---|---|
| **4**<br>**4 5 1 3** | **8** |

**Giải thích:** Hải bốc 3, Hùng bốc 4 hay 1 thì Hải cũng bốc được 5. Số điểm của Hải là 8.

## Bài 48. Gộp đất sét

Có N cục đất sét, viên thứ i có thể tích là $A_i$. Hải chơi một trò chơi gộp đất sét. Mỗi thao tác, Hải có thể gộp hai cục đất xét cạnh nhau lại thành một, chi phí gộp bằng tổng thể tích của hai cục đất này. Hãy nhớ rằng, chỉ được chọn hai cục đất liên tiếp, và ví dụ sau khi gộp cục $A_1$ và $A_2$ thì cục mới này sẽ liên tiếp với $A_3$.

**Yêu cầu:** Hãy lập trình cách chơi tối ưu tốn ít chi phí nhất để Hải gộp N cục đất sét này thành một cục lớn.

**Giới hạn:** N <= 1000 và $A_i <= 10^9$

**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- Dòng tiếp theo chứa N số nguyên dương $A_i$

**Kết quả:** In ra chi phí ít nhất để Hải gộp N cục đất lại. **Ví dụ:**

| input | output |
|---|---|
| **4**<br>**10 20 30 40** | **190** |

**Giải thích:** Ta sẽ gộp theo thứ tự như sau

(10, 20, 30, 40) -> (**30**, 30, 40) tốn 30

(30, 30, 40) -> (**60**, 40) tốn 60

(60, 40) -> (**100**) tốn 100

## Bài 49. Dãy cấp số cộng

Trong toán học, một cấp số cộng (tiếng Anh: arithmetic progression hoặc arithmetic sequence) là một dãy số thoả mãn điều kiện: hai phần tử liên tiếp nhau sai khác nhau một hằng số. Chẳng hạn, dãy số 3, 5, 7, 9, 11,... là một cấp số cộng với các phân tử liên tiếp sai khác nhau hằng số 2. Hằng số sai khác chung được gọi là công sai của cấp số cộng. Các phần tử của nó cũng được gọi là các số hạng.

Bạn cần viết chương trình giải quyết vấn đề sau liên quan đến cấp số cộng: Cho 1 dãy gồm N số nguyên $A_1$ , $A_2$ , ... , $A_n$. Một cấp số cộng là 1 dãy con B của dãy A thoả mãn:

- B[i] = B[i-1] + D
- D là công sai của cấp số cộng này.

**Yêu cầu:** Bạn hãy tìm 1 cấp số cộng dài nhất của dãy A đã cho với công sai là một số D tuỳ ý do bạn chọn thoả mãn $1 < D < 50$
**Giới hạn:** $N <= 2000$ và $A_i <= 10^9$
**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- Dòng tiếp theo chứa N số nguyên dương $A_i$

**Kết quả:** In ra độ dài lớn nhất của dãy con là dãy cấp số cộng.
**Ví dụ:**

| input | output |
|---|---|
| **10**<br>**1 10 3 20 5 30 7 9 40 11** | **6** |

**Giải thích:** Dãy con dài nhất là dãy cấp số cộng : 1, 3, 5, 7, 9, 11 với công sai là 2

## Bài 50. Trò chơi sinh tồn.

Trò chơi sinh tồn là một trò chơi đầy sự hồi hộp cho mọi người. Tài và Tý cùng chơi trò chơi sinh tồn, ban đầu giao kèo rằng: mỗi người có K máu khi bắt đầu trò chơi, ai mất hết K máu (máu = 0) trước sẽ thua. Trò chơi gồm N lá bài được xếp thẳng hàng. Mỗi lá bài có dạng Sự sống hoặc Cái chết. Nếu bốc trúng lá bài Sự sống, người chơi sẽ không mất gì. Ngược lại, bốc trúng lá bài Cái chết, người chơi sẽ bị trừ 1 máu. Tài và Tý thay phiên nhau bốc, mỗi lượt phải bốc một lá bài đầu hàng hoặc cuối hàng. Tài là người chơi đầu tiên.
**Yêu cầu:** Cả N lá bài đều ngửa, ai cũng nhìn thấy, họ sẽ tìm cách chơi tối ưu cho mình. Hỏi ai sẽ giành chiến thắng ?
**Giới hạn:** $N <= 400$ và $K < N$, Dữ liệu đảm bảo có ít nhất 2K - 1 lá bài Cái chết, để đảm bảo rằng có người thua cuộc.
**Dữ liệu vào:**

- Dòng đầu tiên chứa số nguyên dương N
- Dòng tiếp theo chứa N kí tự
    - S có nghĩa là lá bài Sự sống
    - D có nghĩa là lá bài Cái chết

**Kết quả:** In ra TAI nếu Tài sẽ thắng. Ngược lại, in ra TY.
**Ví dụ:**

| input | output |
|---|---|
| **4 1**<br>**SDDD** | **TAI** |
| **6 1 SSDSSD** | **TY** |

**Giải thích:** Ví dụ 1, Tài bốc lá bài S, Tý bốc lá bài nào cũng chết.

# LỜI GIẢI

# Bài 1. Số Fibo

**Lời giải:**

- Bài này là bài làm quen với quy hoạch động
- Muốn tính Q truy vấn với kết quả số fibo thứ N mod cho ($10^9 + 7$), chúng ta không thể mỗi truy vấn, cứ for 1 đến N để tính.
- Ta đã có công thức F[i] = F[i - 1] + F[i - 2]
- Với tính chất (a + b) mod c = (a mod c + b mod c) mod c
- Ta lập mảng F với công thức như sau
  F[i] = (F[i - 1] + F[i - 2]) mod ($10^9 + 7$)

**Cài đặt:**

- Khởi tạo mảng F với công thức đã cho
- Duyệt Q truy vấn, nhập N, in ra F[N]
- Độ phức tạp O(N + Q)

# Bài 2. Tổng tiền tố

**Lời giải:**

- Bài này mỗi truy vấn yêu cầu tính tổng tiền tố kết thúc tại i
- Ta không thể cứ mỗi truy vấn, lại for 1 đến i, làm như vậy sẽ tốn độ phức tạp O(Q * N) , sẽ bị quá thời gian
- Ta sẽ đi khởi tạo mảng F[i] là tổng các số từ $A_1$ đến $A_i$
- Công thức : F[i] = F[i - 1] + A[i]

**Cài đặt:**

- F[0] = 0
- For i từ 1 đến N, gán F[i] = F[i - 1] + A[i]
- For Q truy vấn, nhập k, in ra F[k]

# Bài 3. Chênh lệch độ cao

**Lời giải:**

- Có một thuật toán ngây thơ với độ phức tạp $O(N^2)$ như sau:
  - For j từ 2 đến N, với mỗi j ta lại for i từ 1 đến j - 1 để thử lấy hiệu $H_j$ - $H_i$ rồi cập nhật đáp án.

- Vậy nếu ta cải tiến, ta sẽ giảm bớt 1 vòng for.

  Với mỗi j, nếu ta for i, ta thấy $H_j$ là không đổi. Ta muốn $H_j - H_i$ đạt giá trị lớn nhất. Vậy ta sẽ tìm $H_i$ nhỏ nhất có thể.
- Làm sao để tìm $\min(H_i, H_2, \dots, H_{j-1})$
- Ta sẽ làm tương tự bài 2
- Ta đặt mảng F[i] = min các số $H_1$ tới $H_i$
- Công thức: $F[i] = \min(F[i - 1], H_i)$

**Cài đặt:**

- $F[0] = 10^9$
- For i từ 1 đến N, gán F[i] = min(F[i - 1], H[i])
- For j từ 2 đến N, lấy max tất cả các (H[j] - F[j - 1])

# Bài 4. Tổng đoạn

**Lời giải:**

- Tương tự bài 2
- Ta có F[i] là tổng các số $A_1 + A_2 + \dots + A_i$
- Vậy tổng $A_i + A_{i+1} + \dots + A_j$ sẽ bằng gì ?
- Chính là F[j] - F[i - 1]

**Cài đặt:**

- F[0] = 0
- For i từ 1 đến N, gán F[i] = F[i - 1] + A[i]
- For Q truy vấn, nhập L và R, in ra (F[R] - F[L - 1])

# Bài 5. Đoạn con có tổng lớn nhất.

**Lời giải:**

- Khi ta gọi S[i] là tổng $A_1 + A_2 + \dots + A_i$
- Tổng $A_i + A_{i+1} + \dots + A_j$ bằng S[j] - S[i - 1]
- Vậy ta áp dụng bài 3 (Chênh lệch độ cao)
- Ta có thể tìm được S[j] - S[i] lớn nhất có thể
- Lưu ý bài này có S[0]

**Cách 2:**

- Ta có thể gọi F[i] là tổng đoạn con lớn nhất kết thúc tại i
- Thì có 2 cách tạo thành đoạn con tối ưu
    - Mở ra 1 đoạn mới, F[i] = A[i]
    - Ghép vào đoạn con tối ưu kết thúc là i - 1 thì F[i] = F[i - 1] + A[i]
- Vậy nên ta có công thức: F[i] = max(F[i - 1] , 0) + A[i]
- Khởi tạo: F[0] = 0
- Kết quả bài toán: max tất cả đoạn con kết thúc tại i. Điều này có nghĩa ta phải for i rồi cập nhật kq = max(kq, F[i])

# Bài 6. Hai đoạn con có tổng lớn nhất.

**Lời giải:**

- Ta sẽ tìm hai chỉ số i và j (i < j) sao cho tổng đoạn lớn nhất kết thúc tại i cộng với tổng đoạn lớn nhất bắt đầu tại j là lớn nhất có thể.
- Tổng đoạn lớn nhất kết thúc tại i ta có thể có
    - L[i] = max(L[i - 1], 0) + A[i]
- Tương tự, tổng đoạn lớn nhất bắt đầu tại j là
    - R[j] = max(R[j + 1], 0) + A[i]
- Mảng L ta sẽ khởi tạo trước bằng cách for từ 1 đến N
- Mảng R ta khởi tạo bằng cách for từ N về 1
- Vậy ta cần tìm (i < j) sao cho L[i] + R[j] đạt giá trị lớn nhất.
- Ta sẽ dùng mảng F[i] lưu min(L[1] , L[2] , ... , L[i] )
- Ta for j rồi cập nhật F[j - 1] + R[j] vào đáp án

# Bài 7. Truy vấn tăng đoạn.

**Lời giải:**

- Ta tạo mảng F ban đầu bằng 0
- Khi truy vấn tăng đoạn (L, R) lên K đơn vị
  - Ta gán F[L] = F[L] + K
  - Ta gán F[R + 1] = F[R + 1] - K ?lợi
- Sau khi xong Q truy vấn, ta for i và gán A[i] = A[i - 1] + F[i]
- Mảng A chính là mảng ta cần tìm.

| | | | | | |
|---|---|---|---|---|---|
| An | 0 | 0 | 0 | 0 | 0 |
| f | 7 | 7 | 7 | -7 | -7 |
| f | 7 | 4 | 4 | -4 | -4 |
| | 7 | | | | |
| | | | | | |

# Bài 8. Tổng hình chữ nhật.

**Lời giải:**

- Ta sẽ khởi tạo mảng F[i][j] là tổng của hình chữ nhật có góc trái trên là ô (1, 1) và góc phải dưới là ô (i, j)
- Ta có công thức F[i][j] = F[i][j - 1] + F[i - 1][j] - F[i - 1][j - 1] + A[i][j]
- Đây là tổng tiền tố 2 chiều, các bạn vẽ hình ra sẽ dễ hiểu hơn nhé.
- Và để tính tổng hình chữ nhật có góc trái trên là (x, y) và góc phải dưới là (u, v)
- Ta có F[u][v] - F[x - 1][v] - F[u][y - 1] + F[x - 1][y - 1]

# Bài 9. Truy vấn tăng bảng.

**Lời giải:**

- Cũng tương tự bài truy vấn tăng đoạn, bài này ta cũng tạo mảng F sau đó lấy tổng tiền tố mảng F nhưng là 2 chiều.
- Mỗi truy vấn (x, y, u, v, k)
  - Ta sẽ cộng k cho F[x][y] và F[u + 1][v + 1]
  - Ta trừ k cho F[x][v + 1] và F[u + 1][y]
- Sau đó ta lại có công thức mảng A:
- A[i][j] = A[i - 1][j] + A[i][j - 1] - A[i - 1][j - 1] + F[i][j]
- Mảng A chính là mảng ta cần tìm

# Bài 10. Khối lập phương lớn nhất

**Lời giải:**

- Bài này của thuộc lớp bài mảng cộng dồn
- Nhưng trên hình học không gian 3 chiều
- Nếu bạn đã làm bài 1 chiều và 2 chiều thì có thể làm bài này
- Giả sử đỉnh gốc trục tọa độ là (1, 1, 1)
- Ta cần tính hình lập phương có đỉnh cuối là (x, y, z)
- F[i][j][k] =
  - + F[i -1][j-1][k-1]
  - - (F[i-1][j-1][k]+F[i-1][j][k-1]+F[i][j-1][k-1])
  - + (F[i-1][j][k]+F[i][j-1][k]+F[i][j][k-1])
  - +A[i][j][k]
- Bây giờ ta sẽ for lồng i, j, k thử từng đỉnh cuối (Tốn 3 for)
  - Rồi for t là kích thước cạnh của khối lập phương nhỏ (t <= min(i, j, k) )
    - Thể tích hình ta cần thử sẽ là

- F[i][j][k]
- -F[i-t][j][k]-F[i][j-t][k]-F[i][j][k-t]
- +F[i-t][j-t][k]+F[i-t][j][k-t]+F[i][j-t][k-t]
- -F[i-t][j-t][k-t]

# Bài 11. Biến đổi chuỗi.

**Lời giải:**

- Ta cần đưa về 1 chuỗi mà chữ hoa nằm trước chữ thường
- Vậy ta gọi F[i] là số chữ hoa từ 1 tới i
    - F[0] = 0
    - Nếu kí tự thứ i là in hoa thì F[i] = F[i - 1] + 1
    - Ngược lại, F[i] = F[i - 1]
- Gọi N là chiều dài chuỗi
- Ta sẽ for i = 0 đến N để thử biến đổi toàn bộ kí tự 1 -> i thành chữ in hoa (i = 0 là trường hợp tất cả đều biến thành in thường) và biến đổi toàn bộ kí tự i + 1 -> N làm in thường. (i = N là trường hợp biến tất cả thành in hoa)
- Chi phí :
    - Biến đổi 1 -> i thành in hoa: tốn **(i - F[i])**
    (Đây là số chữ thường trong đoạn 1 -> i)
    - Biến đổi i + 1 -> N thành in thường: tốn **(F[n] - F[i])**
    (Vì đây sẽ là số chữ hoa trong đoạn i + 1 -> N)
- Vậy, ta for i từ 0 đến N, lấy min tất cả (i - F[i] + F[n] - F[i])

# Bài 12. Xếp hàng mua vé

**Lời giải:**

- Gọi F[i] là thời gian nhỏ nhất để i người đầu tiên có được vé
- Ta sẽ chỉ xét người i có mua vé hay không? Vì ta sẽ lấy bài toán con sao cho người i - 1 đã mua vé
    - Nếu người i mua vé, F[i] = F[i - 1] + T[i]

      (F[i - 1] là thời gian i - 1 người đầu tiên đã mua vé, nên ta chỉ việc + thời gian người i mua vé)
    - Nếu người i nhờ người i - 1 mua vé, F[i] = F[i - 2] + R[i - 1] (Lúc này ta không quan tâm i - 2 người đầu tiên mua vé như thế nào, nó đã tối ưu ở F[i - 2])
- Vậy ta có quy hoạch động như sau:
    - F[0] = 0
    - F[1] = T[1]
    - Với i >= 2, F[i] = min(F[i - 1] + T[i], F[i - 2] + R[i - 1])
- Kết quả bài toán: F[n]

# Bài 13. Dãy con tăng dài nhất

**Lời giải:**

- Ta gọi F[i] là dãy con tăng dài nhất **kết thúc** tại phần tử thứ i
- Tất nhiên có một trường hợp, dãy đó chỉ có phần tử
    - F[i] = 1
- Hoặc ta sẽ tìm một vị trí j < i, mà A[j] < A[i]
    - Ta sẽ nối A[i] và dãy đấy

    - F[i] = F[j] + 1
- Vậy ta cài đặt như sau
    - For i từ 1 đến N
        - Cho F[i] = 1
        - For j từ 1 đến i - 1

          Nếu A[j] < A[i] , ta gán F[i] = max(F[i], F[j] + 1) //Điều này có nghĩa là ta cập nhật kết quả tối ưu cho F[i]
- Kết quả bài toán: max tất cả F[i]
- Vì sao kết quả bài toán như vậy ? Các bạn nhớ rằng: F[i] là dãy con tăng dài nhất **kết thúc** tại i, vậy ta phải thử hết tất cả vị trí kết thúc.

## Bài 14. Trò chơi leo thang

**Lời giải:**

- Ta gọi F[i] là số cách đi đến bậc thứ i
- Dĩ nhiên, F[1] = 1
- Nếu bậc 2 bị hỏng => F[2] = 0
    - Ngược lại, F[2] = 1
- Ta for i từ 3 đến N để tính
    - Nếu bậc i bị hỏng, F[i] = 0
    - Có những cách nào để đến bậc thứ i ?
        - Đi từ bậc i - 1 đến: số cách là F[i - 1]
        - Đi từ bậc i - 2 đến: số cách là F[i - 2]
    - Vậy công thức là: F[i] = (F[i - 1] + F[i - 2]) mod ($10^9$ + 7)
    - Vì dữ liệu lớn nên đề yêu cầu ta mod

# Bài 15. Bài toán cái túi

**Lời giải:**

- Việc đầu tiên ta tư duy về bài toán này, là đồ vật thứ i nên chọn hay không ? Nhưng ta không thể duyệt đệ quy $2^N$ được.
- Bây giờ, ta giả sử có 1 phương án tối ưu cho bài toán i đồ vật đầu tiên, sức chứa tối đa là j. Gọi đó là F[i][j]
- Nếu ta có thể tính được F[n][m], đó chính là kết quả bài toán
- Vậy để tính F[i][j], ta sẽ thử nên chọn đồ vật thứ i hay không?
    - Nếu ta không chọn đồ vật thứ i, tất nhiên F[i][j] = F[i - 1][j] (Vì lúc này đồ vật i coi như không xuất hiện, không khác gì xét i - 1 đồ vật đầu tiên)
    - Nếu ta chọn đồ vật thứ i, ta sẽ bỏ vào cái túi sức chứa j đó
        - Vậy ta sẽ lấy bài toán nào để + thêm V[i] vào?
        - Đó chính là bài toán F[i - 1][j - W[i]]
- Vậy ta có công thức F[i][j] như sau
    - F[i][j] = F[i - 1][j] với mọi j
    - Nếu j >= W[i], ta mới có thể bỏ đồ vật i vào túi
        - F[i][j] = max(F[i][j], F[i - 1][j - W[i]] + V[i])
- Kết quả: F[n][m]

# Bài 16. Xâu con chung liên tiếp dài nhất

**Lời giải:**

- Bài này có điều kiện xâu con gồm các kí tự liên tiếp.
- Gọi F[i][j] là xâu con chung liên tiếp dài nhất kết thúc tại i ở xâu A, và kết thúc tại j ở xâu B
- Vì sao bài này lại đặt là kết thúc tại ?
- Vì chúng ta cần là liên tiếp, phải có kết thúc, rồi nối tiếp vào. Chứ

không thể lấy một xâu bất kì đằng trước rồi nối vào được.

- Để tính F[i][j], ta xét 2 trường hợp:
- Nếu $A_i = B_j$ , tất nhiên F[i][j] = F[i - 1][j - 1] + 1
- Nếu $A_i$ khác $B_j$ , F[i][j] = 0

  Vì sao F[i][j] = 0, vốn dĩ muốn xâu con liên tiếp kết thúc tại i bằng xâu con liên tiếp kết thúc tại j, thì tối thiể u ta phải có $A_i = A_j$.
- Khởi tạo: tất cả F[i][j] = 0

# Bài 17. Xâu con chung không liên tiếp dài nhất.

**Lời giải:**

- Bài này là xâu con không cần liên tiếp. Nghĩa là có thể có, có thể không.

Ta gọi F[i][j] là độ dài xâu con chung dài nhất của xâu AiẢ2...Ai và xâu B1B2...BJ

Ví dụ ta có hai xâu

AUXBC

AZBKC

Tất nhiên nhìn vào ta biết F[5][5] = 3, F[4][4] = 2, F[4][3] = 2

Nhưng làm sao để ta tính F[i][j] ?

- Nếu $A_i = B_j$ , ta sẽ lấy luôn kí tự này làm kí tự kết thúc cho xâu con chung ta cần. Vậy F[i][j] = F[i - 1][j - 1] + 1
- Nếu $A_i$ khác $B_j$ , ta lấy F[i][j] = max(F[i - 1][j] , F[i][j - 1])

  Vì trong trường hợp này, xâu con tối ưu chắc chắn là một trong hai trường hợp sau đây:

  - xâu con chung của $A_1A_2...A_i$ và xâu $B_1B_2...B_{j-1}$
  - xâu con chung của $A_1A_2...A_{i-1}$ và xâu $B_1B_2...B_j$

# Bài 18. Dãy con tăng có tổng lớn nhất

**Lời giải:**

- Tương tự bài dãy con tăng dài nhất
- Ta gọi F[i] là tổng lớn nhất của dãy con tăng kết thúc tại i
- Vậy ta sẽ thử j < i mà A[j] < A[i], ta nối A[i] vào
- Thì sẽ là F[j] + A[i]

**Cài đặt:**

- For i từ 1 tới N
    - F[i] = A[i]
    - For j từ 1 tới i - 1
        - Nếu A[j] < A[i] , ta gán F[i] = max(F[i], F[j] + A[i])
- Kết quả: max tất cả F[i]

# Bài 19. Dãy con thân thiện

**Lời giải:**

- Lại một bài tương tự, chủ yếu làm các bạn quen.
- Ta gọi F[i] là dãy con thân thiện dài nhất kết thúc tại i
- Vậy ta sẽ thử j < i mà A[j] < A[i] và thêm điều kiện gcd(A[j], A[i]) = 1, ta nối A[i] vào
- Thì sẽ là F[j] + 1

**Cài đặt:**

- For i từ 1 tới N
    - F[i] = 1
    - For j từ 1 tới i - 1
        - Nếu A[j] < A[i] và gcd(A[i], A[j]) = 1
            - ta gán F[i] = max(F[i], F[j] + 1)

- Kết quả: max tất cả F[i]

# Bài 20. Chơi thể thao

**Lời giải:**

- Ta biểu diễn lại bài toán này, ta sẽ tìm một dãy các ngày Hải sẽ tham gia thi đấu sao cho phần tử A sau phải >= phần tử A trước +K
  - Và tổng B của dãy này tối ưu nhất có thể
- Gọi F[i] là số tiền lớn nhất Hải có được khi xét đến giải đấu thứ i và Hải chắc chắn thi đấu giải i
- Ta sẽ thử những ngày đằng trước là j < i mà A[i] - A[j] >= K để cập nhật F[i] = max(F[i], F[j] + B[i])
  - Hoặc Hải sẽ thi đấu giải i là giải đầu tiên: F[i] = B[i]
- Kết quả bài toán : max tất cả F[i]

# Bài 21. Trò chơi điện tử.

**Lời giải:**

- Với một ô i bất kì, có thể đi từ ô j đằng trước tới với điều kiện

  i - j <= K
- Vậy ta gọi, F[i] là tổng điểm lớn nhất khi đứng tại ô i
- Vì ta có thể dừng cuộc chơi mọi lúc, nên kết quả là max tất cả F[i]
- F[0] = 0
- F[i] = max(F[i], F[j] + A[i]) với điều kiện i - j <= K (j chạy từ 0 -> i - 1)

# Bài 22. Sắp xếp bò đực cái.

**Lời giải:**

- Gọi F[i] là số cách sắp i con bò, thỏa điều kiện 2 con đực được chèn ít nhất K bò cái vào giữa
- Nếu i <= k + 1, ta chỉ đặt được <= 1 con bò đực
  - Ta có i vị trí đặt bò đực

- Hoặc không đặt bò đực
- F[i] = i + 1
- Nếu i > k
  - Nếu ta đặt bò cái tại i, thì lấy F[i - 1]
  - Nếu ta đặt bò đực tại i, tất nhiên một dãy K con bò cái phải đặt liền trước đó, ta lấy F[i - k + 1]
  - Vậy công thức là F[i] = F[i - 1] + F[i - k -1]
  - Lưu ý phải thêm mod

# Bài 23. Không bốc hai liên tiếp.

**Lời giải:**

- Gọi F[i] là tổng lớn nhất bốc được khi xét i phần tử đầu tiên, và đảm bảo điều kiện không được bốc 2 phần tử liên tiếp
- Ta xét 2 trường hợp:
  - Không bốc phần tử i, F[i] = F[i - 1]
  - Không bốc phần tử i - 1 , F[i] = F[i - 2] + A[i]

  (Ở đây ta đã dùng phần tử i - 1 ngăn cách, nên có thể bốc A[i])
- Vậy là cài đặt như sau
- F[0]= 0
- F[1]= A[1]
- Vớii >= 2, F[i] =max(F[i - 1], F[i - 2] + A[i])
- Kết quả bài toán là F[n]

# Bài 24. Trò chơi của Bờm.

**Lời giải:**

- Ta dùng mảng B[x] là tổng các phần tử giá trị x:
- For i từ 1 tới N: B[A[i]] = B[A[i]] + A[i]
- Giá trị <= $10_5$, bây giờ mảng B có $10_5$ phần tử
- Khi xóa x, ta phải xóa luôn x-1 và x + 1.

- Vậy nên đây chính là bài toán không chọn hai phần tử liên tiếp.

- Làm như bài 23

## Bài 25. Không bốc ba liên tiếp.

**Lời giải:**

- Gọi F[i] là tổng lớn nhất bốc được khi xét i phần tử đầu tiên, và đảm bảo điều kiện không được bốc 3 phần tử liên tiếp
- Ta xét 3 trường hợp:
    - Không bốc phần tử i, F[i] = F[i - 1]
    - Không bốc phần tử i - 1 , F[i] = F[i - 2] + A[i]
    - Không bốc phần tử i - 2, F[i] = F[i - 3] + A[i - 1] + A[i]

## Bài 26. Không bốc K liên tiếp.

**Lời giải:**

- Qua bài 24 và 25 chắc các bạn cũng biết bài này nên làm gì
- Công thức tổng quát là:
- Nếu i < K thì F[i] = $A_1 + A_2 + ... + A_i$
- Ngược lại : F[i] = max tất cả F[j-1] + ($A_{j+1} + A_{j+2} + ... + A_i$)
    - Với j >= i - K + 1
    - Và j <= i, trường hợp j = i chính là F[i] = F[i - 1]

## Bài 27. Đếm số đường đi

**Lời giải:**

- Gọi F[i][j] là số đường đi từ ô (1, 1) tới ô (i, j)
- Nếu ô (i, j) có bẫy -> F[i][j] = 0
- Ngược lại, có 2 cách đến ô (i, j)
    - Đi từ ô (i - 1, j) tới: lấy số đường đi của bài toán F[i - 1][j]
    - Đi từ ô (i, j - 1) tới: lấy số đường đi của bài toán F[i][j - 1]
- Vậy công thức ta có là F[i][j] = F[i - 1][j] + F[i][j - 1]
- Khởi tạo: F[1][1] = 1

- 2 vòng for và if ô (i, j) không phải ô (1, 1), ta áp dụng công thức.
    - F[i][j] = (F[i - 1][j] + F[i][j - 1]) mod ($10^9 + 7$)
- Kết quả F[n][n]

# Bài 28. Đường đi tối ưu 1

**Lời giải:**

- Tương tự bài 27, nhưng bài này ta phải cài đặt cho tốt
- Gọi F[i][j] là tổng của đường đi lớn nhất từ (1, 1) tới (i, j)
- Khởi tạo:
    - For i từ 0 tới M
        - For j từ 0 tới N gán F[i][j] = $-10^9$
    - F[1][1] = A[1][1]
- Công thức: F[i][j] = max(F[i - 1][j] , F[i][j - 1]) + A[i][j])
    - Với ô (i, j) không phải ô (1, 1)
- Kết quả: F[m][n]

# Bài 29. Phân trang

**Lời giải:**

- Mình biểu diễn lại bài toán:
- Chia dãy đã cho thành các đoạn liên tiếp, sao cho tổng mỗi đoạn <= L và ta cần làm cho max (L - tổng đoạn) đạt giá trị nhỏ nhất
- Ta gọi F[i] là kết quả bài toán với dãy 1 ... i
- Ta sẽ tìm vị trí 0 <= j < i để cắt thử đoạn [j + 1, i]
    - Điều kiện: ($A_{j+1} + A_{j+2} + ... + A_i$) <= L
    - Hệ số phạt tại đoạn [j + 1, i] là L - ($A_{j+1} + A_{j+2} + ... + A_i$)
    - Hệ số phạt cả trang là: max(F[j], L - ($A_{j+1} + A_{j+2} + ... + A_i$) )
    - Ta cần nhỏ nhất, ta tối ưu cho F[i]:
        - F[i] = min(F[i], max(F[j], L - ($A_{j+1} + A_{j+2} + ... + A_i$)))

- Khởi tạo:
    - Tất cả F[i] = $10^9$
    - F[0] = 0

# Bài 30. Dãy con dài nhất có tổng chia hết cho K

**Lời giải:**

- Đầu tiên, ta chỉ quan tâm số dư cho K, nên ta gán $A_i = A_i$ mod K
- Vì dãy con này không quan trọng điều kiện phần tử đứng sau với phần tử đứng trước, chỉ quan tâm tổng có chia hết cho K hay không?
- Ta gọi F[i][j] là dãy con tăng dài nhất của dãy $A_1, A_2, ..., A_i$ mà chia K dư j
    - Nếu ta không chọn phần tử thứ i vào dãy con, F[i][j] = F[i - 1][j]
    - Nếu ta chọn phần tử thứ i vào dãy con,

      F[i][j] = F[i - 1][(j - A[i] + K) mod K)]

      Vì sao lại (j - A[i] + K) mod K ? Vì (j - A[i] + K) mod K + A[i] sẽ chia K dư j. Sỡ dĩ ta +K vì tránh j - A[i] là âm
    - Ta cần dãy con dài nhất, nên ta lấy F[i][j] = max của 2 trường hợp trên
- Khởi tạo tất cả F[i][j] = $-10^9$ (Kể cả i = 0)
- Cho bài toán cơ sở là F[0][0] = 0
- Kết quả: F[n][0]

# Bài 31. Đếm số biểu thức có tổng bằng S

**Lời giải:**

- Ta có thể coi như bài leo bậc thang, ta muốn leo đến bậc thang S. Mỗi bước nhảy có thể nhảy một trong N khoảng cách $A_1$ , $A_2$, ... $A_n$
- Gọi F[i] là số biểu thức tạo thành tổng i
- F[0] = 1

- Ta for i từ 1 đến S
  - For j từ 1 đến N
    - Nếu i >= A[j] thì F[i] = (F[i] + F[i - A[j]]) mod ($10^9$ + 7)

## Bài 32. Dãy bị khuyết

**Lời giải:**

- Ta gọi F[i][j] là số dãy thỏa mãn kết thúc tại phần tử i, có giá trị j
- Khởi tạo:
  - Nếu $A_1 = 0$ , F[1][j] = 1 với mọi j từ 1 -> m
  - Ngược lại F[1][A[1]] = 1
- Công thức:
  - Nếu $A_i = 0$, F[i][j] = F[i - 1][j] + F[i - 1][j - 1] + F[i - 1][j + 1]
  - Ngược lại,

    F[i][A[i]] = F[i - 1][A[i]] + F[i - 1][A[i] - 1] + F[i - 1][A[i] + 1]

**Lời giải:**

- Lưu ý: thêm mod
- Sắp xếp các cặp số tăng dần theo $A_i$
- Gọi F[i] là số đơn nhiều nhất, kết thúc là đơn i
- F[i] có thể = 1
- Ta tìm đơn j < i mà $B_j < A_i$
  - Để thử F[i] = max(F[i], F[j] + 1)

## Bài 34. Dãy con dài nhất hình chữ V

**Lời giải:**

- Ta gọi L[i] là dãy con giảm dài nhất kết thúc tại i, duyệt từ trái qua
- Gọi R[i] là dãy con giảm dài nhất kết thúc tại i, duyệt từ phải qua
- Khởi tạo tương tự bài dãy con tăng
- Kết quả : max(L[i] + R[i] - 1)

## Bài 35. Trò chơi băng số

**Lời giải:**

- Ta cần tìm một dãy con, với tổng cộng trừ xen kẽ là lớn nhất

- Gọi F[i][0] là dãy con của dãy $A_1$, $A_2$, ... $A_i$ có tổng cộng trừ xen kẽ lớn nhất và kết thúc bằng dấu trừ
- Gọi F[i][1] là dãy con của dãy $A_1$, $A_2$, ... $A_i$ có tổng cộng trừ xen kẽ lớn nhất và kết thúc bằng dấu cộng
- Tất nhiên, với mọi i, ta có một trường hợp F[i][1] = A[i]
- Khi xét tới i
    - Nếu ta chọn i vào dãy
        - F[i][0] = F[i - 1][1] - A[i]
        - F[i][1] = F[i - 1][0] + A[i]
    - Nếu ta không chọn i
        - F[i][0] = F[i - 1][0]
        - F[i][1] = F[i - 1][1]
- Kết quả: F[n][1] (tất nhiên sẽ không lấy dấu trừ ở cuối)

# Bài 36. Nối mạng

**Lời giải:**

- Ta gọi F[i] là chi phí ít nhất để nối i + 1 máy đầu tiên
- F[1] = A[1]
- F[2] = A[1] + A[2]
- Với i từ 3 tới N - 1
    - Chắc chắn ta phải nối i với i + 1, nên ta sẽ có chi phí A[i]
    - Ta có 2 trường hợp nối cho máy i
        - Dùng dây nối i-1 với i : F[i] = F[i - 1] + A[i]
        - Không dùng dây nối i-1 với i : F[i] = F[i - 2] + A[i] ○

    Vậy công thức là: F[i] = max(F[i - 1], F[i - 2]) + A[i]
- Kết quả: F[n - 1]

# Bài 37. Cắt bìa

**Lời giải:**

- Gọi F[a][b] là số lát cắt ít nhất để đưa mảnh bìa hình a x b về các mảnh hình vuông
- Ta có b - 1 cách cắt thẳng, i chạy từ 1 tới b - 1
  - F[a][b] = min(F[a][i] + F[a][b - i] + 1)
- Ta có a - 1 cách cắt ngang, i chạy từ 1 tới a - 1
  - F[a][b] = min(F[i][b] + F[a - i][b] + 1)
- Ta lấy min các trường hợp đó lại

# Bài 38. Nhảy cóc 1

**Lời giải:**

- Ta gọi F[i] là chi phí ít nhất để đến được hòn đá i
- Để đến hòn đá i, ta có thể:
  - Nhảy từ i - 1 tới: F[i] =F[i - 1] + $|H_i - H_{i-i}|$
  - Nhảy từ i - 2 tới: F[i] =F[i - 2] + $|H_i - H_{i-2}|$
- Công thức : F[i]= min(F[i] =F[i - 1] + $|H_i - H_{i-1}|$ , F[i - 2] + $|H_i - H_{i-2}|$)
- Trường hợp cơ sở:
  - F[1] = 0
  - F[2] = $|H_1 - H_2|$
- Kết quả: F[n]

# Bài 39. Nhảy cóc 2

**Lời giải:**

- Ta gọi F[i] là chi phí ít nhất để đến được hòn đá i
- Để đến hòn đá i , ta có thể nhảy từ j tới với điều kiện
  - j >= 1

  - $j + K >= i$
  - Chi phí: $F[j] + |H_j - H_i|$
- Ta for i từ 2 tới N
  - Để ý điều kiện, ta chỉ for j tầm K lần
- Độ phức tạp O(N * K)

# Bài 40. Vận động viên ba môn.

**Lời giải:**

- Ta gọi F[i][0] là số tiền lớn nhất Hải nhận được khi tham gia i ngày đầu tiên và ngày thứ i là Bóng đá
- Ta gọi F[i][1] là số tiền lớn nhất Hải nhận được khi tham gia i ngày đầu tiên và ngày thứ i là Cầu lông
- Ta gọi F[i][2] là số tiền lớn nhất Hải nhận được khi tham gia i ngày đầu tiên và ngày thứ i là Điền kinh
- Trường hợp cơ sở:
  - F[1][0] = B[1]
  - F[1][1] = C[1]
  - F[1][2] = D[1]
- Công thức
  - F[i][0] = max(F[i - 1][1],F[i -1][2]) + B[i]
  - F[i][1] = max(F[i - 1][0],F[i -1][2]) + C[i]
  - F[i][2] = max(F[i - 1][0],F[i -1][1]) + D[i]
- Kết quả: max(F[n][0], F[n][1], F[n][2])

# Bài 41. Chia kẹo cho hai bé.

**Lời giải:**

- Ta gọi F[i][j] = true nếu tồn tại dãy con của dãy $A_1, A_2, ..., A_i$ có tổng là j. Ngược lại F[i][j] = false
- Muốn tồn tại dãy con của dãy $A_1, A_2, ..., A_i$ có tổng là j:
  - Trường hợp 1: Tồn tại dãy con của $A_1, A_2, ..., A_{i-1}$ có tổng

j

- F[i - 1][j] là true

  - Trường hợp 2: Tồn tại dãy con của $A_1$, $A_2$, ..., $A_{i-1}$ có tổng

j - A[i]

- F[i - 1][j - A[i]] là true

  - Ta cần thỏa một trong hai trường hợp thì F[i][j] = true

- Ta muốn chia dãy A thành 2 phần sao cho chênh lệch nhỏ nhất
- Gọi X là tổng phần lớn, Y là tổng phần nhỏ
- Ta muốn X - Y nhỏ nhất, mà X + Y = Sum
- => X ta cần tìm là số nhỏ nhất lớn hơn hoặc bằng Sum/2
- Vậy ta duyệt j từ Sum/2 tới Sum, nếu F[n][j] = true, ta sẽ lấy giá trị X đấy: kết quả là |X - (Sum - X)| = 2*X - Sum

# Bài 42. Đổi tiền

**Lời giải:**

- Gọi F[k] là số tờ ít nhất để đổi thỏi vàng giá trị k ra tờ tiền byteland.
- Được hiểu rằng F[k] = $10^9$ (dương vô cùng ) (không thể đổi)
- Khởi tạo:
  - F[0] = 0
  - F[i] = $10^9$ với i >= 1
- Với một thỏi vàng giá trị K bất kì, ta sẽ thử từng loại tờ tiền i từ 1 -> N và cập nhật kết quả tối ưu cho F[k]
  - For k từ 1 tới M
    - For i từ 1 tới N
      - Nếu k >= A[i] thì gán F[k] = min(F[k], F[k - A[i]])
- Kết quả : F[M]

# Bài 43. Dãy con chung dài nhất

**Lời giải:**

- Ta cần tìm dãy con chung dài nhất của M dãy.
- Nhận xét: Dãy ta tìm được sẽ là dãy con của dãy 1
- Vậy ta tìm một dãy con của dãy đầu tiên thỏa mãn điều kiện
    - Với hai số x, y liên tiếp ở dãy con của dãy 1, thì x phải nằm trước y ở các dãy khác
        - Hay ta đánh B[i][x] là vị trí của giá trị x ở hàng i
        - Điều kiện là B[i][x] < B[i][y] với mọi i
- Độ phức tạp $O(N_2 * M)$

# Bài 44. Hình vuông cùng màu lớn nhất

**Lời giải:**

- Ta gọi F[i][j] là độ dài cạnh hình vuông cùng màu lớn nhất có đỉnh phải dưới là ô (i, j)
- Nếu ô (i, j) khác màu 1 trong 3 ô (i - 1, j); (i, j - 1); (i - 1, j - 1) tất nhiên F[i][j] = 1
- Ngược lại, F[i][j] = min(F[i - 1][j], F[i][j - 1], F[i - 1][j - 1]) + 1
- Kết quả: max tất cả F[i][j]

# Bài 45. Thuê xe du lịch

**Lời giải:**

- Gọi F[j] là chi phí nhỏ nhất để chở đúng j người
- Ta sẽ thử thuê xe i và cập nhật F[j] = min(F[j], F[j - a[i]] + b[i])
- Kết quả: min(F[j]) với j >= m và j <= m + 200

# Bài 46. Xây tháp ngoài hành tinh

**Lời giải:**

- Nhận xét: Với 2 loại khối bất kì, loại nào có sức chịu A nhỏ hơn ta sẽ luôn xếp ở dưới

- Vậy đầu tiên, ta phải sort N loại lại theo thứ tự tăng dần sức chịu A. Mục đích là để ta có thể quy hoạch động, cập nhật lên.
- Gọi F[s] = true nếu có thể xây dựng được độ cao s
- Dĩ nhiên F[0] = true
- Cách cài đặt:
- For i từ 1 tới N
    - For s từ A[i] về H[i]
        - For k từ 1 tới C[i]
        - Nếu F[s - k *H[i]] thỏa mãn true thì F[s] cũng gán = true
- Đáp án: For s từ giới hạn 40000 về 0, nếu F[s] = true thì thỏa mãn, in ra độ cao s và kết thúc.

# Bài 47. Bốc sỏi

**Lời giải:**

- Đầu tiên ta có phương pháp đệ quy
- F ( L, R ) là số điểm lớn nhất của người bốc đầu trong đoạn [L, R]
- Trường hợp người đầu bốc L
    - F(L, R) = tổng đoạn [L, R] - F(L + 1, R)
- Trường hợp người đầu bốc R
    - F(L, R) = tổng đoạn [L, R] - F(L, R - 1)
- Vậy công thức tổng quát là
    - F(L, R) = tổng đoạn [L, R] - min( F(L+1,R) , F(L, R - 1))
- Ta cài theo kiểu đệ quy có nhớ.
- Kết quả là F(1, n)

# Bài 48. Gộp đất sét

**Lời giải:**

- Bài này ta Quy Hoạch Động theo cách đệ quy có nhớ

- Bạn phải làm được đệ quy cho bài này
- Sau đó lưu F[L][R] là chi phí nhỏ nhất để gộp đoạn [L, R] thành 1 cục
- Vậy ta phải for j từ L tới R-1 để thứ cắt đoạn [L, R] thành 2 . Hay nói cách khác là gộp đoạn [L, j] và đoạn [j+1, R] lại
- Công thức
- F[L][R] = min(F[L][R], dequy(L, j) + dequy(j + 1, R))
- Kết quả: F[1][n]

## Bài 49. Dãy cấp số cộng

**Lời giải:**

- Bài này ta lưu ý giới hạn D <= 50
- Ta for thử giá trị D
    - Tìm dãy con cấp số cộng dài nhất tăng theo D
    - Làm tương tự bài dãy con tăng
- Kết quả : max tất cả các F[i] ở tất cả giá trị D mình duyệt

## Bài 50. Trò chơi sinh tồn.

**Lời giải:**

- Bài này đầu tiên ta phải đệ quy quay lui được
- Rồi đánh dấu lại (đệ quy có nhớ) F[i][j][a][b] là true/false nếu người chơi đầu thắng trong đoạn [i, j] và a, b là số máu còn lại.
- Sau đó với nhận xét, mỗi giá trị a, có một giá trị b tương ứng, ta không cần lưu chiều b nữa.