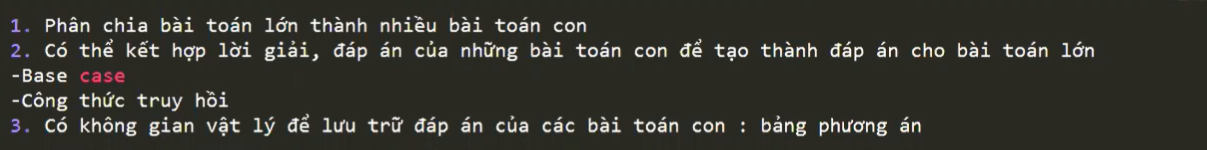
Quy Hoạch Động:



B1. Phân rã

B2. Giải các phân rã

B3. Tổng hợp

1. Một số dạng bài:
2. Bài 1: Dãy con tăng dài nhất (LIS)

Cho mảng số nguyên A[] gồm N phần tử, hãy tìm dãy con (không nhất thiết các phần tử phải liên tiếp) tăng chặt dài nhất của dãy A.

Input: N và số phần tử ( 1<=N<=1000; 1<=A[i]<=1000)

Output: Độ dài của dãy con dài nhất

Sample:

* 14
* 128 104 53 876 660 961 754 775 220 231 224 915 312 994
* 6

1. Bài 2: Dãy con tổng bằng S

Cho mảng số nguyên A[] gồm N phần tử và số nguyên dương S, hãy xác định xem có thể tạo ra 1 tập con các phần tử trong mảng có tổng bằng S hay không?

Input: N và S, N phần tử ( 1 <=N<=200; 1<=S<=60000; 1<=A[i]<=300)

Output: In ra 1 nếu có, ngược lại in ra 0

* 8 92
* 69 16 82 170 31 24 45 112
* 1

1. Bài 3: Coin problem

Ngân hàng XYZ hiện tại có N tờ tiền có mệnh giá khác nhau được được lưu vào mảng C[], bạn hãy tìm số cách đổi số tiền là S sao cho số tiền cần dùng là ít nhất. Bạn được sử dụng 1 mệnh giá vô hạn lần

Input: N,S và các giá trị C[N] (1<=N<=100; 1<=S<=1e6; 1<=C[i]<=1e6)

Output: In ra số tiền nhỏ nhất cần đổi. Nếu không được thì in ra -1

* 3 10
* 4 5 8
* 2

1. Bài 4: Coin 2

Hãy xem xét 1 hệ thống tiền tệ của ngân hàng XYZ bao gồm n đồng xu. Mỗi đồng xu có một giá trị nguyên dương. Nhiễm vụ của bạn là tính số cách riêng biệt mà bạn có thể tạo ra số tiền x bằng cách sử dụng số xu có sẵn. Ví dụ: Nếu số xu là {2,3,5} và tổng mong muốn là 9. Có 8 cách 2 + 2 + 5, 2 + 5 + 2, 5 + 2 + 2, etc

Input: n và x, giá trị của các đồng xu ( 1<=n<=100; 1<=x<=1e6; 1<=c[i]<=1e6)

Output: In ra kết quả lấy dư mod

* 3 9
* 2 3 5
* 8

1. Bài 5: Coin3:

Hãy xem xét một hệ thống tiền tệ của ngân hàng ABC gồm n đồng xu. Mỗi đồng xu có giá trị nguyên dương. Nhiệm vụ của bạn là tính số cách riêng biệt không xét đến thứ tự mà bạn có thể tạo ra số tiền X bằng những đồng xu có sẵn . Ví dụ nếu số xu là {2,3,5} và tổng mong muốn là 9 thì có 3 cách : 2+2+5; 3+3+3; 2+2+2+3

Input: n và X, n số phần tử C[] (1<=n<=100; 1<=X<=1e6; 1<=c[i]<=1e6)

Output: In kết quả lấy dư mod

* 3 9
* 2 3 5
* 3

1. Bài 6: Bài toán cái túi (Kapsnack)

Một người có cái túi thể tích V ( V<1000). Anh ta có N đồ vật cần mang theo (N<=1000), mỗi đồ vật có thể tích A[i] (A[i] <= 100) và giá trị là C[i] ( C[i] <=100). Hãy xác định tổng giá trị lớn nhất của các đồ vật mà người đó có thể mang theo, sao cho tổng thể tích không vượt quá V.

Input: N, V. N số giá trị mảng A. N số giá trị mảng C ( V<1000, N<=1000, A[i] <=100, C[i] <=100)

Output: In ra giá trị lớn nhất có thể đạt được

* 6 22
* 39 44 4 59 91 70
* 47 26 92 33 6 69
* 92

1. Bài 7: Dãy con có tổng S

Cho mảng số nguyên A[] gồm N phần tử và số nguyên dương S, hãy xác định xem có thể tạo ra một tập con các phần tử trong mảng có tổng bằng S không.

Input: N,S, N số phần phần tử A (1<=N<=200; 1<=S<=6000; 1<=A[i]<=300)

Output: In ra 1 nếu có tập con của A có tổng bằng S, ngược lại in in ra 0

* 8 92
* 69 16 82 170 31 24 45 112
* 1

1. Bài 8: Xâu con chung dài nhất (LCS)

Cho hai xâu kí tự S và T, hãy tìm xâu con chung dài nhất của S và T. Các kí tự của xâu con không nhất thiết phải liền kề nhau.

Input: S, T (length <=1000)

Output: In ra độ dài xâu con chung dài nhất của S và T

* ZHFTDFHF
* TFISHROV
* 3

1. Bài 9: Tổng không liền kề

Cho mảng A[] gồm N phần tử, nhiệm vụ của bạn là tính tổng lớn nhất trong dãy con trong mảng với một điều kiện đó là trong dãy con này không được có hai phần tử nằm liền kề nhau

Input: N, N phần tử trong mảng A (1<=N<=1e6; 1<=A[i]<=1000)

Output: In ra kết quả của bài toán

* 7
* 314 514 148 451 896 589 296
* 1706

1. Bài 10: Số bước ít nhất

Cho mảng A[] gồm N số nguyên. Nhiệm vụ của bạn là sắp xếp lại mảng với số lượng bước là ít nhất. Tại mỗi bước bạn chỉ được phép chèn phần tử bất kì của mảng vào vị trí bất kì trong mảng. Ví dụ A[]= {2,3,5,1,4,7,6} sẽ cho ta số phép chèn ít nhất là 3 bằng cách lấy số 1 chèn trước số 2, lấy số 4 chèn trước số 5 và số 6 chèn trước số 7, ta nhận được mảng đã được sắp xếp

Input: N, N phần tử của mảng A (1<=N<=1000; 1<=A[i]<=1000)

Output: Đưa ra kết quả trên 1 dòng

* 13
* 143 340 571 845 221 228 278 443 666 594 491 814 24
* 6

1. Bài 11: Staircase

Một chiếc cầu thang có N bậc. Mỗi bước bạn chỉ được phép bước lên tối đa K bước. Hỏi có tất cả bao nhiêu cách bước để đi hết cầu thang? (Tổng số bước N)

Input: N và K (1<=N<=100000; 1<=K<=100)

Output: In ra đáp án lấy dư mod

* 7 3
* 44

1. Bài 12: Frog SPOJ

Một con ếch có thể nhảy 1,2,3 bước để có thể lên đến một đỉnh cần đến. Hãy đếm số cách con ếch có thể nhảy đến đỉnh.

Input: Một số nguyên N miêu tả số bước cần ( 1<=N<=40)

Output: In ra kết quả.

* 5
* 13

1. Bài 13: Xem phim SPOJ

John có một đàn bò. Một ngày đẹp trời, anh ta quyết định mua xe tải với khả năng chở được C kg (1000<=C<= 25000) để đưa những con bò đi xem phim. Cho số con bò N(20<=N<=100) và khối lượng w[i] của từng con đều nhỏ hơn C. Hãy cho biết khối lượng bò lớn nhất mà John có thể đưa đi xem phim là bao nhiêu

Input: C và N, N số phần tử của w[] ( 1000<=C<=25000; 20<=N<=100; w[i]<=C)

Output: Tổng khối lượng bò lớn nhất mà John có thể mang đi xem phim

* 259 5
* 81 58 42 33 61
* 242

1. Bài 14: Số nguyên lớn

Cho hai số nguyên N và M không quá 1000 chữ số. Người ta muốn tính xem liệu có thể lấy ra nhiều nhất bao nhiêu chữ số trong N (Không cần liên tiếp) và giữ nguyên thứ tự của nó để tạo ra 1 số X sao cho ta cũng thể tìm thấy X trong số M theo cách tương tự

Input: N và M (length <=1000)

Output: In ra nhiều chữ số nhất của X

* 82619136359
* 5572555993152891122
* 5

1. Bài 15: Tổng lớn nhất của dãy con tăng dần

Cho dãy số A[] gồm N số. Nhiệm vụ của bạn là tính tổng lớn nhất của dãy con được sắp xếp tăng dần của dãy A[]. Ví dụ với A[] = {1,101,2,3,100,4,5} ta có kết quả là 106 = 1 + 2 + 3 + 10. Với dãy A[] ={ 10,7,5} ta có kết quả là 10. Với dãy A[]={1,2,3,5} ta có kết quả là 11

Input: N, N phần tử của A (1<=N<=1000; 0<=A[i]<=1000)

Output: Đưa ra kết quả

* 8
* 2 12 3 15 3 16 11 4
* 45

1. Bài 16: Dãy Bitonic

Một dãy số được gọi là Bitonic nếu nó được chia thành hai dãy đầu tăng dần và dãy sau giảm dần. Nhiệm vụ của bạn là tìm tổng lớn nhất dãy con Bitonic của dãy số A[]. Ví dụ: A[]={1,15,51,45,33,100,12,18,9} ta có kết quả là 194 tương ứng với dãy Bitonic {1,15,51,100,18,9}

Input: N và N phần tử của dãy A (1<=N<=100; 1<=A[i]<=100)

Output: Đưa ra kết quả

* 8
* 7 8 8 19 3 6 2 15
* 49

1. Bài 17: Xúc xắc

Tèo hiện đang học sinh lớp 10A1, vốn là một học sinh được đánh giá rất cao về khả năng toán học nên Tèo thường được các bạn trong lớp nhờ giải các bài toán trên tạp chí toán học và tuổi trẻ. Hôm nay Tí là bạn thân của Tèo gặp một bài toán trên tạp chí và đã nhờ Tèo giải hộ, Tèo dù rất giỏi toán và biết cách làm bài này nhưng bài toán này kết quả quá lớn nên Tèo phải nhờ tới bạn lập trình thuật toán để tìm ra lời giải giúp Tèo.

Bài toán có yêu cầu rất đơn giản, cho ngẫu nhiên số N, hỏi rằng nếu được tung một con xúc xắc không giới hạn số lần và lấy tổng số chấm trên mặt xúc xắc cộng lại với nhau, hỏi có bao nhiêu cách tung để tổng số chấm trong mọi lần tung xúc xắc bằng N. Bạn hãy giúp Tèo giải quyết bài toán này nhé, vì số cách tung có thể quá lớn, nên bạn hãy chia dư kết quả với modulo 10^9 + 7.

Input: Dòng duy nhất chứa số nguyên dương N ( 1<=N<=1e6)

Output: Kết quả của bài toán

* 7
* 63

1. Bài 18: Maximun Path Sum

Cho bảng A[] kích thướt M x N ( N hàng, M cột). Bạn được đi xuống dưới, đi sang phải và đi xuống ô chéo dưới về bên phải. Khi đi qua ô (i,j), điểm nhận được bằng A[i][j]. Hãy tìm đường đi từ ô (1,1) đến ô (N,M) sao cho tổng điểm là lớn nhất.

Input: N và M, giá trị của A[i][j]

Output: In kết quả bài toán

* 5 5
* 1 0 31 5 25
* 28 26 32 46 7
* 26 40 36 13 16
* 7 26 14 6 11
* 42 45 11 10 21
* 208

1. Bài 19: Đếm số đường đi trong mê cung

Cho mê cung được miêu tả bởi ma trận N hàng N cột. Mỗi ô trên ma trận chứa kí tự ‘.’ tương ứng với đường đi và ‘\*’ tương ứng với bẫy. Một con chuột muốn di chuyển từ ô (1,1) đến ô (N,N) và chỉ được di chuyển khi một ô nào đó là đường đi và nó được di chuyển sang các ô chung cạnh với ô nó đang đứng hiện tại. Hãy đếm số cách con chuột có thể di chuyển tới đích.

Input: N, vả N kí tự

Output: Số đường đi tối đa lấy dư mod

* 4
* . . . .
* . \* . .
* . . . \*
* \* . . .
* 3

1. Bài 20: Maximun Square SPOJ

Bạn được cung cấp một ma trận nhị phân có N dòng và M cột chỉ bao gồm các số 0 và 1, hãy tìm hình vuông lớn nhất trong ma trận chỉ chứa toàn số 1

Input: N, M và ma trận ( 1<=N,M<=500; 0<=A[i][j] <=1)

Output: In đáp án

* 4 4
* 1 1 0 0
* 1 1 1 1
* 1 1 0 1
* 1 0 1 0
* 2

1. Bài 21: