1. **Cài đặt thuật toán 8 con hậu.**

Hướng dẫn:

Chạy file Tamhau.py.

Input:

Output: Cách đặt con hậu trên bàn cờ:

Ma trận 8\*8

Mỗi hàng là gồm 7 số 0 và 1 số 1 là ví trị đặt con hậu.

Mã giả:

Try (i)

for j =1 to 8 do

if not ok\_row(i) and not ok\_col(j) and not ok\_plus(i+j-1) and not ok\_minus(i-j+8) then

solution(i) = j

ok\_col(j)=true

ok\_plus(i+j-1)=true

if i<8 then

try(i+1)

else print\_solution()

ok\_row(j) = false

ok\_col(j) = false

ok\_plus(i+j-1) = false

ok\_minus(i-j+8) = false

1. **Cài đặt bài toán tháp HaNoi**

Hướng dẫn:

Chạy file ThapHaNoi.py

Nhập n là số tự nhiên (Cho biết số lượng đĩa) từ bàn phím.

Input: số tự nhiên n.

Output: gồm nhiều dòng là các bước xếp đĩa.

Thuật toán: Dùng đệ quy

Đầu tiên ta lấy cột C làm cọc trung gian. Chuyển n-1 chiếc đĩa sang cột B.

Ta chuyển chiếc đĩa lớn nhất sang cột C.

Lấy cột A làm trung gian chuyển n-1 chiếc đĩa từ cột B sang cột C.

1. **Cài đặt bài toán balo 1**

Hướng dẫn

Chạy file Balo1.py

Dữ liệu được nhập từ bàn phím.

Input: Dòng thứ nhất chứa 2 số nguyên n, s (trong đó: n là số đồ vật, s là trọng lượng tối đa của cái túi.

Dòng thứ 2: chứa n số nguyên dương là trọng lượng của mỗi đồ vật.

Dòng thứ 3: chứa n số nguyên dương là giá trị của mỗi đồ vật.

Output: In ra giá trị tối đa của đồ vật mà cái túi chứa đựa.

Thuật toán: Sử dụng giải thuật quy hoạch động

Ta có dp[i][j]: Giá trị lớn nhất của cái túi khi lựa chọn các đồ vật 1,2,3,4..i và có trọng lượng là j.

Nếu i=0 hoặc j=0, thì dp[i][j] bằng 0

Nếu w[i] > j, dp[i][j] bằng dp[i-1][j] //không lấy

Nếu w[i]<=j, dp[i][j] bằng max (dp[i-1][j],dp[i-1][j-w[i]]+v[i]): Nếu lựa chọn đồ vật thứ I cho vào túi.

1. **Cài đặt bài toán balo 2**

Hướng dẫn

Chạy file Balo1.py

Dữ liệu được nhập từ bàn phím.

Input: Dòng thứ nhất chứa 2 số nguyên n, s (trong đó: n là số đồ vật, s là trọng lượng tối đa của cái túi.

Dòng thứ 2: chứa n số nguyên dương là trọng lượng của mỗi đồ vật.

Dòng thứ 3: chứa n số nguyên dương là giá trị của mỗi đồ vật.

Output: In ra giá trị tối đa của đồ vật mà cái túi chứa đựa.

Thuật toán:

Ta có dp[i][j]: Giá trị lớn nhất của cái túi khi lựa chọn các đồ vật 1,2,3,4..i và có trọng lượng là j.

Nếu i=0 hoặc j=0, thì dp[i][j] bằng 0

Nếu w[i] > j, dp[i][j] bằng dp[i-1][j] //không lấy

Nếu w[i]<=j, dp[i][j] bằng max (dp[i-1][j],dp[i][j-w[i]]+v[i]): Nếu lựa chọn đồ vật thứ I cho vào túi.

1. **Cài đặt bài toán đổi tiền**

Hướng dẫn.

Chạy file DoiTien.py

Dữ liệu được nhập từ bàn phím.

Input: Số tự nhiên n, s trong đó n là số mệnh giá, s là số tiền cần đổi

Dãy gồm n phần tử là mệnh giá cần đổi.

Output: Số tờ tiền ít nhất cần đổi.

Vd:

Input:

13

1 5 6 9

Output: 3 (6 6 1)

Thuật toán: Sử dụng giải thuật quy hoạch động.

Công thức QHĐ:

F[I,j] với I là mệnh giá tiền, j là số tiền cần đổi

F[I,0] =0 // Nếu số tiền đổi là 0

F[0,j]=inf // Nếu mệnh giá là 0

F[i][j]=min(f[i-1,j],f[I,j-c[i]]+1)

1. **Cài đặt bài toán Mã đi tuần**

Hướng dẫn:

Chạy file Madituan.py

Dữ liệu nhập từ bàn phím.

Input: x, y (0<=x,y<8)là vị trí đầu tiên con mã trên bàn cờ.

Output: Ma trận 8x8. Mỗi A[i][j] là thứ tự nước đi của con mã.

Thuật toán:

-Xây dựng nước đi cho con mã:

Gọi x, y là độ dài bước đị trên các trục Oxy. Một bước đi hợp lệ của quân mã sẽ như sau: |x|+ |y| = 3 (Với x, y >0).

Khi đó ở một ví trí trí bất kì quân mã có có 8 đường có thể di chuyển. Chưa xét đến bước đi đó có hợp lệ hay không.

Các bước đi đó là: ( -2, -1), ( -2, 1), ( -1, -2), ( -1, 2), ( 1, -2), ( 1, 2), ( 2, -1), ( 2, 1)

Để đơn giản ta sẽ tạo hay mảng X[], Y[] để chứa các giá trị trên. Với mỗi X[i], Y[i] sẽ là một cách di chuyển của quân mã(0 ≤i≤ 7 ).

-Kiểm tra tính hợp lệ của bước đi

Ta sẽ dùng một mảng hai chiều A[n\*n] để lưu vị trí của từng ô trong bàn cờ. Tất cả mảng đều khởi tạo giá trị là 0 (quân mã chưa đi qua).

Giọi x, y là vị trí hiện tại của quân mã, thì vị trí tiếp theo mà quân mã đi sẽ có dạng x+X[i], y+Y[i]. Một vị trí được gọi là hợp lệ thì sẽ thỏa mãn tính chất sau:

* 0 ≤ x+X[i]≤ n-1.
* 0 ≤ x+X[i]≤ n-1.

Nếu bước đi đó là bước đi đúng thì ta sẽ lưu thứ tự của bước đi đó vào mảng A[ x+X[i], y+Y[i] ].

1. **Cài đặt quiksort**

Hướng dẫn:

Chạy file quiksort.py

Dữ liệu nhập từ bàn phím.

Nhập n số nguyên là phần tử của mảng, cách nhau bởi khoảng trắng.

Input:

Dãy số gồm n số nguyên, mỗi phần tử cách nhau bởi khoảng trắng.

Output.

Dãy n số nguyên đã được sắp xếp từ bé đến lớn.

Thuật toán:

Bước 1: Nếu left>=right//dãy có ít hơn 2 phần tử

Kết thúc;

Bước 2: phân hoạch hãy aleft…aright thành các đoạn: aleft…aj,aj+1…ai-1,ai…aright

Đoạn 1: aleft…aj<=mid

Đoạn 2: aj+1…ai-1=mid

Đoạn 3: ai…aright>=mid

Với mid là giá trị bất kì trong mảng. Thông thường là giá trị tại vị trí chính giữa của mảng.

* Bước 3: Sắp xếp đoạn 1: aleft…aj

Cách làm như đoạn ban đầu.

* Bước 4: Sắp xếp đoạn 3: ai…aright

Cách làm như đoạn ban đầu.

1. Tìm chuỗi chung dài nhất.

Hướng dẫn

Chạy file LCS.py

Dữ liệu nhập từ bàn phím.

Intput: Xâu X

Xâu Y

Output: Độ dài xâu con dài nhất.

Thuật toán:

F[i][j]: Lưu độ dài sau con chung dài nhất có được bằng cách lấy i kí tự đầu tiên của xâu X, và j kí tự đầu tiên cảu xâu Y.

F[n][m]: Đáp án của bài toán.

Cơ sở quy hoạch động.

F[i][0]=0

F[0][i]=0

X[i] và Y[j]:

X[i]=Y[j]=>F[i][j]=F[i-1][j-1]+1

X[i]!=Y[j] => F[i][j]=max(F[i][j-1],F[i-1][j]).