**HƯỚNG DẪN GIẢI BÀI PY01075 – TRÒ CHƠI TRÊN ĐƯỜNG THẲNG**

1. **Cơ sở toán học**

***Bổ đề Bezout.***

Nếu d là ước số chung lớn nhất của hai số nguyên a và b thì sẽ tồn tại hai số nguyên x và y sao cho d=ax+by.

***Mở rộng cho nhiều số***

Nếu d là UCLN (a1, a2, …, an) thì luôn tồn tại bộ số (x1, x2, …, xn) sao cho:

a1\* x1 + a2\* x2 + … + an \* xn = d

1. **Lời giải bài toán**

Giả sử tập các bước nhảy các thẻ bạn Nam sử dụng là (x1, x2, …, xn). Sau nhiều lần di chuyển, giả sử toạ độ của Nam bây giờ là d thì:

a1\* x1 + a2 \* x2 + … + an \* xn = d

Yêu cầu bài toán là phải chọn tổ hợp làm sao để đi được đến mọi điểm. Nghĩa là tổ hợp (x1, x2, …) được chọn **CỐ ĐỊNH**  thì phải luôn có nghiệm với mọi d bất kỳ được chọn

Gọi P là UCLN của tập số ta chọn. Theo định lí Bê – du, ta chứng minh được luôn tồn tại bộ các số (a1, a2, …, an) sao cho:

a1\* x1 + a2 \* x2 + … + an \* xn = P

Nhân cả 2 vế phương trình với số d khác 0, ta có:

da1 \* x1 + da2 \* x2 + … + d\*an \* xn = Pd

* Rút ra các điều sau:
* Bộ nghiệm của phương trình là (da1, da2, …)
* Phương trình này chỉ luôn đáp ứng được với mọi điểm có toạ độ Pd. Tức là các điểm có toạ độ là bội số của UCLN của tập số đã chọn. Các điểm có toạ độ không phải Pd chưa chắc đã di chuyển được

***Như vậy, để với tổ hợp bước nhảy được chọn, mà luôn có nghiệm với mọi điểm thì buộc UCLN của tập đó chỉ có thể là 1***

***Ta chứng minh***

Theo định lí Bê – du, ta có luôn tồn tại bộ số (a1, a2, …, an) sao cho:

a1\* x1 + a2 \* x2 + … + an \* xn = 1

Nhân cả 2 vế của phương trình với số nguyên d khác 0, ta được

da1 \* x1 + da2 \* x2 + … + d\*an \* xn = d

Rõ ràng, với mọi điểm có toạ độ d luôn xác định được bộ các số (da1, da2, …, d\*an)

**KẾT LUẬN: Nội dung của bài toán chính là:**

|  |
| --- |
| ***Hãy tìm tập các thẻ thoả mãn 2 điều kiện: Ước chung lớn nhất của tất cả bước nhảy là 1, đồng thời tổng chi phí là nhỏ nhất có thể*** |

**CÁCH GIẢI: Duyệt vét cạn**

Ta sẽ sử dụng một map với key là UCLN của tập bước nhảy, và value là giá thành nhỏ nhất trong tất cả các tập hợp số được chọn có chung UCLN đó

from math import \*

for \_ in range(int(input())):

n = int(input())

l = list(map(int, input().split()))

c = list(map(int, input().split()))

gcd\_cost = {0: 0} #key là GCD, value là chi phí tối thiểu để đạt GCD đó

# Duyệt qua từng thẻ

for i in range(n):

#Mỗi thẻ ta sẽ thử thêm vào các tập thẻ hiện có

current\_costs = list(gcd\_cost.items())

for g, cost in current\_costs:

new\_g = gcd(g, l[i])

new\_cost = cost + c[i]

if new\_g not in gcd\_cost or new\_cost < gcd\_cost[new\_g]: gcd\_cost[new\_g] = new\_cost

if 1 in gcd\_cost: print(gcd\_cost[1])

else: print(-1)

**GIẢI BÀI ĐẢO DẤU**

Cho dãy số A có n phần tử và một số k. Bạn phải thực hiện k phép biến đổi. Mỗi phép biến đổi, ta phải lấy một số và thay nó bằng số đối. Hãy thực hiện đủ k lần biến đổi sao cho tổng các phần tử sau cùng trong dãy là bé nhất có thể:

**CÁCH GIẢI**

* Sắp xếp mảng tăng dần
* Ta sẽ ưu tiên đổi các số âm có giá trị bé trước cho đến khi hết lượt đổi dấu được phép
* Nếu đã hết lượt đổi dấu nhưng vẫn còn số âm thì thôi, không làm gì nữa
* Nếu đổi dấu hết số âm rồi, trong dãy toàn số dương nhưng vẫn còn thừa lượt phải thực hiện thì ta xét tính chẵn lẻ:

+ Nếu số lượt dư ra là chẵn: Để bảo toàn tất cả các số là dương thì chọn đại một số dương rồi cứ thực hiện: đảo dấu lần 1 – thành âm, đảo dấu lần 2 – thành số dương

+ Nếu số lượt dư ra là lẻ: Để bảo toàn nhiều số dương nhất có thể: Để bảo toàn tất cả các số là dương thì chọn đại một số dương rồi cứ thực hiện: đảo dấu lần 1 – thành âm, đảo dấu lần 2 – thành số dương

***Lúc này còn dư ra 1 phép biến đổi phải thực hiện: Dĩ nhiên ta sẽ chọn số dương có giá trị thấp nhất để đảo thành dấu âm***