# 復習済み

A-E

# A 9x9

正解

# B tcaF

正解

# C Snake Queue

正解

考え方も同じ

# D Squares in Circle

正解

paizaでもやった。

# E Square Price

不正解

N種類、10^100のすべての商品の値段がわかるから、それらから安い順に選んで行って合計がM以下までの個数を見る。これが基本的な考え方。ただ、前から(安いやつから)足していくのは効率が悪いので、その高速化として二分探索によってその境界値の商品の値段を求めている。

重要なポイントとして、赤い文字のところのコード

if amount > M:

break

が途中離脱のコードだが、ないとあるテストケースでTLEを起こす。

この問題の最悪のパターンは

N = 2\*10^5

M = 10^18

Pがすべて1の時である。

実際、ないと2200ms以上でTLEで、あると111msである。

自分のパソコンでも4.27sが1.76sと劇的に変化する。

でも正直、これをしなきゃいけないという感覚が持てない。

## 正解コード

N, M = map(int, input().split())

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

ok = 0

ng = 10\*\*18+1

amount = 0

while abs(ng-ok) > 1:

x = (ok+ng)//2

amount = 0

for i in range(N):

k = (x+P[i])//(2\*P[i])

amount += k\*\*2 \* P[i]

if amount > M:

break

if amount <= M:

ok = x

else:

ng = x

amount = 0

ans = 0

for i in range(N):

k = int( (ok+P[i])/(2\*P[i]) )

amount += k\*\*2 \* P[i]

ans += k

for i in range(N):

if M-amount >= ok+1:

if (ok+1)%P[i] == 0 and ((ok+1)//P[i]+1)%2 == 0:

amount += ok+1

ans += 1

else:

break

print(ans)

# F Rated Range

解いてない

# G Odd Even Graph

解いてない