

Бонбони и балони

locked

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Асистентът ви много харесва балони и иска да му давате балони всеки ден в продължение на N дни (номерирани с числата от 1 до N). На поредния ден i, вашият асистент иска Ai балона. Проблемът е, че имате само M балона.

За щастие може да давате бонбони вместо балони на асистента си. На поредния ден i той е съгласен да получи Bi бонбона за всеки балон който не сте му дали, или по-формално казано, ако му дадете Xi балона на ден i, тогава трябва да му дадете и $C_i = \max(0, A_i - X_i) \times B_i$ бонбона.

Задачата ви е да минимизирате максималният брой бонбони, които трябва да дадете на асистента си в един ден – намерете минималната възможна стойност на $\max(C_1, C_2, ..., C_N)$.

Input Format

На първият ред от входа ще са дадени числата N и M.

На вторият ред от входа ще бъдат дадени числата A1, A2, ..., AN.

На третият ред от входа ще бъдат дадени числата B1, B2, ..., BN.

Constraints

$1 \leq N \leq 10^5$

$N \leq M \leq 10^{18}$

$0 \leq A_i \leq 10^9$

$0 \leq B_i \leq 10^9$

Output Format

Изведете едно число - минималната стойност на $\max(C_1, C_2, ..., C_N)$.

Sample Input 0

```
5 3
1 2 3 4 5
1 2 3 4 5
```

Sample Output 0

```
15
```

Explanation 0

Оптималното разпределение на балоните по дни е (0, 0, 0, 1, 2) което води и до отговора 15 (бонбоните които ще трябва да дадете на 5тия ден).

(Ограничението $N \leq M$ е изпълнено във всички останали тестове, тук M е по-млако за по-лесно разписване на теста)

Sample Input 1

```
5 6
1 3 3 3 2
4 1 5 3 7
```

Sample Output 1

```
5
```

Explanation 1

Оптималното разпределение на балоните по дни е (0, 0, 2, 2, 2) което води и до отговора 5 (бонбоните които ще трябва да дадете на 3тия ден).

Current Buffer (saved locally, editable) C++

```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 using namespace std;
7 bool isSweetsEnough(vector<pair<unsigned long long int,unsigned long long int>> vec, long double minSweets,int
days,unsigned long long int balloons){
8     unsigned long long int counter=0;
9     for(int i=0;i<days;i++){
10         if(vec[i].second==minSweets){
11             counter+=ceil((vec[i].second-minSweets)/vec[i].first);
12         }
13         if(counter>balloons){
14             return 0;
15         }
16     }
17     return 1;
18 }
19 long long int binarySearch(vector<pair<unsigned long long int,unsigned long long int>> vec,unsigned long long int
right,int days,unsigned long long int balloons){
20     if(balloons==0){
21         return right;
22     }
23     unsigned long long int left=0;
24     unsigned long long int answer;
25     while(left<right){
26         long long int mid=(left+right)/2;
27         if(isSweetsEnough(vec,mid,days,balloons)){
28             right=mid;
29             answer=mid;
30         }
31         else{
32             left=mid+1;
33         }
34     }
35     return answer;
36 }
37 int main() {
38     int days;
39     cin>>days;
40
41     unsigned long long int balloons;
42     cin>>balloons;
43
44     vector<pair<unsigned long long int,unsigned long long int>> vec;
45     unsigned long long int* firstArr=new unsigned long long int[days];
46     unsigned long long int* secondArr=new unsigned long long int[days];
47     pair<unsigned long long int,unsigned long long int> p;
48     unsigned long long int right=0;
49
50     for(int i=0;i<days;i++){
51         cin>>firstArr[i];
52     }
53     for(int i=0;i<days;i++){
54         cin>>secondArr[i];
55     }
56     for(int i=0;i<days;i++){
57         p.first=secondArr[i];
58         p.second=firstArr[i]*secondArr[i];
59         vec.push_back(p);
60         if(p.second>right){
61             right=p.second;
62         }
63     }
64     cout<<binarySearch(vec,right,days,balloons);
65     return 0;
66 }
67
```

Line: 14 Col: 11

Upload Code as File

Test against custom input

Run Code

Submit Code