CSES Problem Set

Factory Machines

Submission details

Task:	Factory Machines
Sender:	razs
Submission time:	2024-12-10 18:52:54 +0200
Language:	Python3 (PyPy3)
Status:	READY
Result:	ACCEPTED

Test results ▲

test	verdict	time	
#1	ACCEPTED	0.04 s	»
#2	ACCEPTED	0.04 s	»
#3	ACCEPTED	0.04 s	»
#4	ACCEPTED	0.04 s	»
#5	ACCEPTED	0.04 s	»
#6	ACCEPTED	0.05 s	»
#7	ACCEPTED	0.05 s	»
#8	ACCEPTED	0.21 s	»
#9	ACCEPTED	0.17 s	»
#10	ACCEPTED	0.29 s	»
#11	ACCEPTED	0.04 s	»
#12	ACCEPTED	0.04 s	»
#13	ACCEPTED	0.05 s	»
#14	ACCEPTED	0.04 s	»
#15	ACCEPTED	0.04 s	»
#16	ACCEPTED	0.36 s	»
#17	ACCEPTED	0.04 s	»
#18	ACCEPTED	0.04 s	»
#19	ACCEPTED	0.21 s	»
#20	ACCEPTED	0.04 s	»

Code ▲

```python
def minimális_idő(n, t, gépek):
    # Bináris keresés határai
    bal = 1
    jobb = t * min(gépek)
    válasz = jobb

    while bal <= jobb:
        közép = (bal + jobb) // 2

        # Számoljuk meg, hány termék készül el adott idő alatt
        összes_termék = sum(közép // k for k in gépek)

        if összes_termék >= t:
            # Ha elég termék készül, csökkentjük az időtartamot
            válasz = közép
            jobb = közép - 1
        else:
            # Ha nem elég, növeljük az időtartamot
            bal = közép + 1

    return válasz

# Bemenet olvasása
import sys
bemenet = sys.stdin.read
adatok = bemenet().splitlines()

n, t = map(int, adatok[0].split())
gépek = list(map(int, adatok[1].split()))

# Minimális idő kiszámítása
eredmény = minimális_idő(n, t, gépek)

# Kimenet
print(eredmény)
```

SHARE CODE TO OTHERS

Your submissions

2024-12-10 18:52:54

Test details

Test 1

Verdict: ACCEPTED

input
```
10 10
6 5 1 2 1 5 10 4 6 6
```

correct output
```
4
```

user output
```
4
```

Test 2

Verdict: ACCEPTED

input
```
10 10
6 6 4 3 4 9 3 2 6 10
```

correct output
```
6
```

user output
```
6
```

Test 3

Verdict: ACCEPTED

input
```
10 10
5 4 10 7 8 4 1 8 9 2
```

correct output
```
5
```

user output
```
5
```

Test 4

Verdict: ACCEPTED

input
```
1 1000000000
1
```

correct output
```
1000000000
```

user output
```
1000000000
```

Test 5

Verdict: ACCEPTED

input
```
1 1000000000
1000000000
```

correct output
```
1000000000000000000
```

user output
```
1000000000000000000
```

Test 6

Verdict: ACCEPTED

input
```
1000 1000
271 687 392 992 11 410 702 870...
```

correct output
```
223
```

user output
```
223
```

Test 7

Verdict: ACCEPTED

input
```
1000 1000
598 523 703 794 737 689 724 26...
```

correct output
```
282
```

user output
```
282
```

Test 8

Verdict: ACCEPTED

input
```
200000 1000000000
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
```

correct output
```
5000
```

user output
```
5000
```

Test 9

Verdict: ACCEPTED

input
```
200000 1
760045594 599341056 300098860 ...
```

correct output
```
8214
```

user output
```
8214
```

Test 10

Verdict: ACCEPTED

input
```
200000 1000000000
33041040 210038922 596070148 7...
```

correct output
```
371045814100
```

user output
```
371045814100
```

Test 11

Verdict: ACCEPTED

input
```
25 1000000000
1000000000 1 1 1 1 1 1 1 1 1 1...
```

correct output
```
41666667
```

user output
```
41666667
```

Test 12

Verdict: ACCEPTED

input
```
12 1000000000
1 1 1 1 1 1 1 1 1 1 1 10000000...
```

correct output
```
90909091
```

user output
```
90909091
```

Test 13

Verdict: ACCEPTED

input
```
23 1000000000
1000000000 1000000000 10000000...
```

correct output
```
434782610000000000
```

user output
```
434782610000000000
```

Test 14

Verdict: ACCEPTED

input
```
3 3
10 11 12
```

correct output
```
12
```

user output
```
12
```

Test 15

Verdict: ACCEPTED

input
```
43 1000000000
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
```

correct output
```
27105055
```

user output
```
27105055
```

Test 16

Verdict: ACCEPTED

input
```
200000 1000000000
999990001 999991200 999991130 ...
```

correct output
```
5000000000000
```

user output
```
5000000000000
```

Test 16

Verdict: ACCEPTED

input
```
200000 1000000000
999990001 999991200 999991130 ...
```

correct output
```
5000000000000
```

user output
```
5000000000000
```

Test 17

Verdict: ACCEPTED

input
```
1 1
1
```

correct output
```
1
```

user output
```
1
```

Test 18

Verdict: ACCEPTED

input
```
100 1000000000
1000000000 1000000000 10000000...
```

correct output
```
10000000000000000
```

user output
```
10000000000000000
```

Test 19

Verdict: ACCEPTED

input
```
200000 1000000000
2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
```

correct output
```
5001
```

user output
```
5001
```

Test 19

Verdict: ACCEPTED

input
```
200000 1000000000
2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
```

correct output
```
5001
```

user output
```
5001
```

Test 20

Verdict: ACCEPTED

input
```
2 1000000000
2 3
```

correct output
```
1200000000
```

user output
```
1200000000
```