



实验舱
青少年编程
走近科学 走进名校

蛟龙五班

二分答案、浮点数的二分

Mas

二分答案

一个区间中的左侧或右侧都满足某种条件,而另一侧都不满足条件,也可以看作是一种有序(把满足条件看做 1,不满足看做 0)

二分可以用来查找满足某种条件的最大/小的值

不难想到从小到大枚举这个作为答案的最大值,然后判断是否合法,若答案单调可以使用二分查找快速找到答案

要想使用二分来解这种**最大值最小化**的题目,需要满足以下条件:

- 答案在一个固定区间内
- 要求能比较容易地判断某个值是否符合条件的
- 可行解对于区间满足一定的单调性

最小值最大化同理



#1336 木材加工

题目描述

木材厂有一些原木,现在想把这些木头切割成一些长度相同的小段木头(木头可能有剩余),需要得到的小段的数目是给定的

当然,我们希望得到的小段木头越长越好,你的任务是计算能够得到的小段木头的最大长度

木头长度的单位是 cm 。原木的长度都是正整数,我们要求切割得到的小段木头的长度也是正整数

例如有两根原木长度分别为 11 和 21,要求切割成到等长的 6 段,很明显能切割出来的小段木头长度最长为 5

输入格式

第一行是两个正整数 N 和 K ($1 \leq N \leq 100000, 1 \leq K \leq 100000000$), N 是原木的数目, K 是需要得到的小段的数目

接下来的 N 行,每行有一个 1 到 100000000 之间的正整数,表示一根原木的长度

输出格式

能够切割得到的小段的最大长度。如果连 1 cm 长的小段都切不出来,输出 0

输入样例

3 7

232

124

456

输出样例

114

#1336 木材加工

从最小值~最大值枚举时间复杂度为 $O(nk)$

从二分思想去枚举切割的长度

只需要遍历一次输出即可求出能切除的段数

记录最后一次合法的值即为答案

时间复杂度为 $O(n\log k)$

枚举的长度	1	2	3	4	5	6	7	8	9	10
能切除的段数	40	19	12	8	6	5	4	3	2	1
是否满足条件	是	是	是	是	是	否	否	否	否	否

```
#include <bits/stdc++.h>
using namespace std;
int a[100001], n, m, l, r, mid, ans;
bool check(int len)
{
    int res = 0;
    for (int i = 0; i < n; i++)
        res += a[i] / len;
    return res >= m;
}
int main()
{
    cin >> n >> m;
    for (int i = 0; i < n; i++)
        cin >> a[i], r = max(r, a[i]);
    l = 0;
    while (l <= r)
    {
        mid = (l + r) >> 1;
        if (check(mid))
        {
            ans = mid;
            l = mid + 1;
        }
        else
            r = mid - 1;
    }
    cout << ans << endl;
    return 0;
}
```

#1319、砍树

题目描述

伐木工人米尔科需要砍倒 M 米长的木材

米尔科只被允许砍倒单行树木

米尔科的伐木机工作过程如下：米尔科设置一个高度参数 H (米),伐木机升起一个巨大的锯片到高度 H ,并锯掉所有的树比 H 高的部分(当然,树木不高于 H 米的部分保持不变)。米尔科就得到树木被锯下的部分。

例如,如果一行树的高度分别为 20, 15, 10 和 17 ,米尔科把锯片升到 15 米的高度,切割后树木剩下的高度将是 15 , 15 , 10 和 15 ,而米尔科将从第 1 棵树得到 5 米,从第 4 棵树得到 2 米,共得到 7 米木材

米尔科非常关注生态保护,所以他不会砍掉过多的木材

帮助米尔科找到伐木机锯片的最大的整数高度 H ,使得他能得到木材至少为 M 米

换句话说,如果再升高 1 米,则他将得不到 M 米木材

输入格式

第一行输入两个整数 N 和 M , N 表示树木的数量和需要的木材总长度

第二行输入 N 个整数表示每棵树的高度,值均不超过 1000000000

所有木材长度之和大于 M ,因此必有解

输出格式

输出一个整数,表示砍树的最高高度

随着高度升高,获得的木材数量将减少

二分枚举高度 H

对于每个高度,检查是否满足条件

输入样例

```
5 20
4 42 40 26 46
```

输出样例

```
36
```

数据规模

对于全部的数据 $1 \leq N \leq 1000000, 1 \leq M \leq 2000000000$

#265、魔法数字

题目描述

一个正整数是魔法数字当且仅当它是 A 的倍数或者它是 B 的倍数。

给出 A 和 B ，求第 n 小的魔法数字。

数据输入

一行三个数字，分别表示 A ， B 和 n 。

数据输出

一个数字，第 n 小的魔法数字。

范围说明

对于 50% 的数据有： $1 \leq A, B, n \leq 100$ 。

对于 100% 的数据有： $1 \leq A, B \leq 10^4, 1 \leq n \leq 10^9$ 。

样例输入1

```
1 5 10
```

样例输出1

```
10
```

样例输入2

```
2 3 4
```

样例输出2

```
6
```

#265、魔法数字

对于一个数 x

通过简单容斥不难发现 x 以内有 $\frac{x}{A} + \frac{x}{B} - \frac{x}{[A,B]}$ 个魔法数

魔法数的编号显然具有单调性

可以直接二分枚举时间复杂度可将答案的右边界设为 10^{18}

至多进行 59 次枚举，每次枚举检查答案时间复杂度 $O(1)$

可以思考数学做法

```
#include <bits/stdc++.h>
using namespace std;
long long l = 1, r = 1e18, a, b, ans, n, t;
bool check(long long x)
{
    return (x / a + x / b - x / t) >= n;
}
int main()
{
    cin >> a >> b >> n;
    t = a / __gcd(a, b) * b;
    while (l <= r)
    {
        long long mid = l + r >> 1;
        if (check(mid))
        {
            ans = mid;
            r = mid - 1;
        }
        else
        {
            l = mid + 1;
        }
    }
    cout << ans;
    return 0;
}
```

#1264、奶牛晒衣服

题目描述

洗完衣服后，你就要弄干衣服。衣服在自然条件下用 1 的时间可以晒干 A 点湿度。抠门的熊大妈买了 1 台烘衣机。

使用烘衣机可以让你用 1 的时间使 1 件衣服除了自然晒干 A 点湿度外，还可以烘干 B 点湿度，但在 1 的时间内只能对 1 件衣服使用。

N 件衣服因为种种原因而不一样湿，现在告诉你每件衣服的湿度，要你求出弄干所有衣服的最少时间（湿度为 0 为干）。

输入格式

第一行 N, A, B

接下来 N 行，每行一个数，表示衣服的湿度（ $1 \leq \text{湿度}$ ， $A, B \leq 500000, 1 \leq N \leq 500000$ ）

输出格式

一行，弄干所有衣服的最少时间。

输入样例

```
3 2 1
1
2
3
```

输出样例

```
1
```

说明

第 1 个时间内，用机器处理第 3 件衣服，此外，所有衣服自然晒干。花费 1 时间全部弄干。

#1264、奶牛晒衣服

随着时间的增加,衣服更容易自然晾干

二分枚举时间 x

需要使用烘干机的时间为

$$\sum_{i=1}^n \max(0, \left\lceil \frac{a_i - A \times x}{b} \right\rceil)$$

是否有其他思路?

当有多件衣服未干时,给湿度最大的衣服进行烘干显然更优

记当前使用时间为 x

当所有衣服中湿度最大值不超过 $A \times x$ 时,说明剩余衣服可自然晾干

可使用优先队列维护上述过程

```
bool check(Long Long s)
{
    Long Long res = 0;
    for (int i = 1; i <= n; i++)
    {
        if (t[i] <= a * s)
            continue;
        res += ceil(1.0 * (t[i] - a * s) / b);
    }
    return res <= s;
}
```

#1253、时间管理

题目描述

作为一名忙碌的商人,约翰知道必须高效地安排他的时间

他有 N 工作要做,比如给奶牛挤奶,清洗牛棚,修理栅栏之类的

为了高效,约翰列出了所有工作的清单.第 i 份工作需 T_i 单位的时间来完成,且必须在 S_i 或之前完成

现在是 0 时刻,约翰做一份工作必须直到做完才能停止

所有的商人都喜欢睡懒觉

请帮约翰计算他最迟什么时候开始工作,可以让所有工作按时完成.(如果无法完成全部任务,输出 -1)

输入格式

第一行一个整数 N

第 $2 \sim N + 1$ 行,每行输入两个整数 T_i, S_i

输出格式

约翰最晚开始工作的时间,如果不能完成工作请输出 -1

输入样例

```
4
3 5
8 14
5 20
1 16
```

输出样例

```
2
```

数据规模

对于全部的数据 $1 \leq n \leq 100000, 1 \leq T_i \leq 10^9, 1 \leq S_i \leq 10^9$

#1253、时间管理

随着开始时间的后移,越不容易完成所有任务

二分枚举开始时间 t

对所有任务按结束时间排序

记前 i 个任务的完成时的时间为 x ,初始时令 $x = t$

每次累加当前任务的耗时

若发现 x 当前任务的截至时间,说明在该开始时间下无法完成

是否有其他思路?

#622、数列分段 II

题目描述

对于给定的一个长度为 N 的正整数数列 A ，现要将其分成 M 段，并要求每段连续，且每段和的最大值最小。

例如，将数列 $4\ 2\ 4\ 5\ 1$ 要分成 3 段：

若分为 $[4\ 2][4\ 5][1]$ ，各段的和分别为 $6, 9, 1$ ，和的最大值为 9 ；

若分为 $[4][2\ 4][5\ 1]$ ，各段的和分别为 $4, 6, 6$ ，和的最大值为 6 ；

并且无论如何分段，最大值不会小于 6 。

所以可以得到要将数列 $4\ 2\ 4\ 5\ 1$ 要分成 3 段，每段和的最大值最小为 6 。

输入格式

第 1 行包含两个正整数 N, M ；

第 2 行包含 N 个空格隔开的非负整数 A_i ，含义如题目所述。

输出格式

仅包含一个正整数，即每段和最大值最小为多少。

样例输入

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

样例输出

```
6
```

数据范围与提示

对于 20% 的数据，有 $N \leq 10$ ；

对于 40% 的数据，有 $N \leq 1000$ ；

对于 100% 的数据，有 $N \leq 10^5$ ， $M \leq N$ ， A_i 之和不超过 10^9 。

#622、数列分段 II

二分枚举这个最大值 x ,将划分段数不小于 M 作为条件

随着 x 的减小,段数相应增多越容易满足条件

对于每一段划分参考 [#621数列分段](#) 进行限制为 x 的最少段数的划分

如果划分的最少段数不超过 M ,那么一定可以分成 M 段

计算出划分的最少段数

相当于求解01序列中1第一次出现的位置



#1854、数的三次方根

题目描述

给定一个浮点数 n ，求它的三次方根。

输入格式

共一行，包含一个浮点数 n 。

输出格式

共一行，包含一个浮点数，表示问题的解。

注意，结果保留 6 位小数。

输入样例：

```
1000.00
```

输出样例：

```
10.000000
```

数据范围

对于全部数据 $-10000 \leq n \leq 10000$

不是整数域的二分!?

对于浮点数的二分 $|l - r| < \text{精度}$, 可直接视作区间为空

对于 l, r 不能直接加减 1, 而应该直接加减最小精度(直接赋值为 mid 也可, 为什么?)

直接二分三次方根

```
#include <bits/stdc++.h>
using namespace std;
double x, l = -10000, r = 10000, mid, ans;
int main()
{
    cin >> x;
    while (r - l >= 0.00000001)
    {
        mid = (l + r) / 2;
        if (mid * mid * mid >= x)
        {
            ans = mid;
            r = mid;
        }
        else
            l = mid;
    }
    printf("%.6lf", ans);
    return 0;
}
```

#2083、月饼

题目描述

中秋节要到了！根据习俗， Mas 需要将一些月饼分给大家。

他有 N 个不同口味、不同大小的月饼。有 F 个朋友,每个人会拿到一块月饼（必须一个月饼的一块，不能由几个月饼的小块拼成；可以是一整个月饼）。

Mas 朋友们都特别小气，如果有人拿到更大的一块，就会开始抱怨。因此所有人拿到的月饼是同样大小的（但不需要是同样形状的），虽然这样有些月饼会被浪费，但总比搞砸整个月饼对好。当然， Mas 也要给自己留一块，而这一块也要和其他人的同样大小。

请问每个人拿到的月饼最大是多少？每个月饼都是一个高为 1，半径不等的圆柱体。

输入格式

第一行包含两个正整数 N 和 F ， $1 \leq N \leq F \leq 10000$ ，表示月饼的数量和朋友的数量。

第二行包含 N 个 $1 \sim 10000$ 之间的整数，表示每个月饼的半径。

输出格式

输出每个人能得到的最大的月饼的体积，精确到小数点后五位。

样例输入

```
3 3
4 3 3
```

样例输出

```
25.13274
```

$$\pi = 3.141592653589793$$

$$V = \pi \times r^2 \times h$$

#2083、月饼

二分枚举每个小朋友获得的月饼体积 x

对于每一块月饼能平均分成 $\left\lfloor \frac{v_i}{x} \right\rfloor$ 块,一共能平分成 $\sum_{i=1}^n \left\lfloor \frac{v_i}{x} \right\rfloor$ 块

将块数不小于 $F + 1$ 作为条件

```
#include <bits/stdc++.h>
#define PI 3.141592653589793
using namespace std;
double a[10001], l = 0, r, mid, ans;
int n, f;
bool check(double x)
{
    int cnt = 0;
    for (int i = 0; i < n; i++)
        cnt += (int)(a[i] / x);
    return cnt >= f + 1;
}
int main()
{
    cin >> n >> f;
    for (int i = 0; i < n; i++)
        cin >> a[i], a[i] = PI * a[i] * a[i], r = max(r, a[i]);
    for (int i = 0; i < 1000; i++)
    {
        mid = (l + r) / 2;
        if (check(mid))
        {
            ans = mid;
            l = mid;
        }
        else
            r = mid;
    }
    printf("%.5lf", ans);
    return 0;
}
```


#2083、月饼

实数二分若想结束二分过程,除了判断 $r - l$ 是否落到指定精度里,还可以直接指定总次数

显然这样能更好地计算总复杂度

二分总次数可以根据 $r - l$ 的最大范围和精度要求来估算

在本题中,我们取 $l = 0, r = \max(\pi a_i^2) \leq 3.14 \cdot 10^8$

精度要求是小数点后五位,所以二分的总位数 $\sim 10^{13}$

注意到 $2^{50} \geq 10^{15}$,所以二分次数设成50次即可

一般 100 次能满足几乎所有题目的精度要求

当然在时间允许下,可考虑多增加一些循环次数

#2076、银行贷款

题目描述

Mas 从银行贷款了 1000 元钱，月利率为 10%，三个月内分三期还款，那他每次要还款 402.11 元。银行告诉他是这么算的：

$$\frac{402.11}{1.1} + \frac{402.11}{1.1^2} + \frac{402.11}{1.1^3} \approx 1000$$

现在已知总贷款数 x 元，每个月还贷 y 元，一共贷款 n 个月，要求计算出贷款者向银行支付的利率，假设利率按月累计。

输入格式

三个用空格隔开的正整数 x, y, n ($1 \leq x, y \leq 10^5, 1 \leq n \leq 48$)，分别表示总贷款数，每月还贷数目，一共的月份。

输出格式

一个实数，表示该贷款的月利率(用百分数表示)，四舍五入精确到 0.01%。答案保证不超过 1000%。

样例输入

```
1000 100 12
```

样例输出

```
2.92
```

#2076、银行贷款

当利率增大时还款总数将增大

二分枚举每个月的利率 p

对于每个利率 p ,计算其还款总数

判断是否超过总数 x



实验舱
青少年编程
走近科学 走进名校

谢谢观看