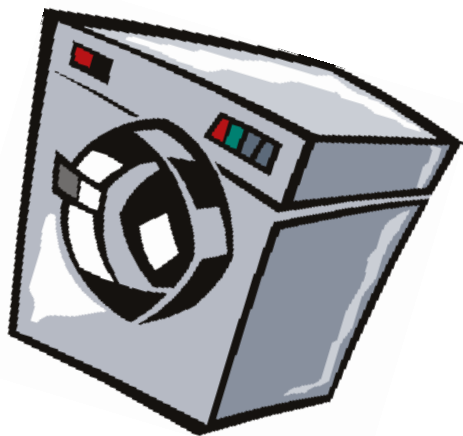




# 二分与贪心算法

郭 炜 刘家瑛

# POJ 3104 烘晾衣服





# 问题描述

- 现有  $n$  件衣服需要烘干
- 每件衣服的含水量为  $a_i$ 
  - 如果自然晾干, 每分钟含水量减少 1
  - 如果使用烘干机烘干, 每分钟含水量减少  $k$  (直至为0)
- 只有一台烘干机, 每次只能烘干一件衣服
- 且一次至少使用1分钟
- 求使所有衣服含水量为0的最少时间是多少



## ■ 程序输入

- 输入包含三行
  - 第一行是一个整数  $n$  ( $1 \leq n \leq 100000$ ),  
表示衣服的数量
  - 第二行有  $n$  个整数,  
分别表示各件衣服的含水量  $a_i$  ( $1 \leq a_i \leq 10^9$ )
  - 第三行是一个整数  $k$  ( $1 \leq k \leq 10^9$ ),  
表示烘干机1分钟减少的水量



## ▀ 程序输出

- 输出一行, 表示最少需要的时间



## ■ 样例输入

3

2 3 9

5

## ■ 样例输出

3



# 解题思路

- 要求最小的时间

- 问题转换:

最小值问题  $\rightarrow$  判定性问题

- 判断在时间 $X$ 内, 是否能晾干/烘干所有的衣服

- 很明显, 这个问题满足单调性的条件

$\rightarrow$  对时间 $X$ 进行二分来解决!



# 解题思路

- 对于给定的时间 $X$ , 依次判断每一件衣服  $i$ 
  - 如果  $a_i \leq X$ , 则该衣服可以自然烘干
  - 否则说明需要烘干机,
    - 因为多用 1分钟 烘干机, 可以多减少  $(k-1)$  的水量
    - 至少需要  $\text{ceil}((X - a_i) / (k-1))$  分钟的烘干机
- 如果 所有衣服需要烘干机的时间总和  $\leq X$   
则说明时间 $X$ 是可行的, 否则说明不可行





# 重点函数分析

## 判断X是否可行

```
bool check(X) {  
    当前需要烘干机时间 = 0;  
    for ( 所有的衣服 ) {  
        if ( 第i件衣服含水量 > x ) {  
            当前总计需要烘干机时间 += 第i件衣服至少需要的烘干机时间;  
        }  
    }  
    //判断最后的烘干机时间是否小于等于x  
    return 当前需要烘干机时间 <= x  
}
```



# 参考程序

```
#include <stdio.h>
#include <iostream>
using namespace std;

#define MAXN (100000+10)

int n; //n件衣服
int l, r, mid; //用于判定时间x, 所考虑的左右及中值
int k; //烘干机1分钟减少的水量
int a[MAXN];
```



# 参考程序

//判断时间为ans是否可行

```
bool check(int ans) {  
    int now = 0; //需要烘干机的时间  
    for (int i = 0; i < n; ++i )  
        if (a[i] > ans){  
            now += (a[i]-ans-1)/(k-1) + 1;  
            //第i件衣服需要烘干机的时间,注意是(k-1)  
            //  $[a/k] = (a-1)/k + 1$   
            if (now > ans) return false;  
        }  
    return true;  
}
```



# 参考程序

```
int main() {  
    scanf("%d", &n);  
    l = 0; r = 0;  
    for (int i = 0; i < n; ++i) {  
        scanf("%d", &a[i]);  
        if (a[i] > r) r = a[i];  
    }  
    scanf("%d", &k);  
    if (k == 1) printf("%d\n", r);  
    //k=1, 直接输出, 避免除0的情况
```



# 参考程序

```
else {  
    //二分答案,判断可行性  
    while (l <= r) {  
        mid = (l+r)/2;  
        if (check(mid)) r = mid-1;  
        else l = mid+1;  
    }  
    printf("%d\n", l);  
}  
return 0;  
} //end main
```