实验舱
青少年编程
走近科学 走进名校

蛟龙四班

简单贪心

Mas

贪心

贪心算法（*greedy algorithm*）,是用计算机来模拟一个“贪心”的人做出决策的过程

并不是所有的时候贪心法都能获得最优解,所以一般使用贪心法的时候,都要确保自己能证明其正确性

适用范围

贪心算法在有最优子结构的问题中尤为有效

最优子结构是指问题能够分解成子问题来解决,子问题的最优解能递推到最终问题的最优解

证明方法

- 反证法

  如果交换方案中任意两个元素/相邻的两个元素后,答案不会变得更好,那么可以推定目前的解已经是最优解了

- 归纳法

  先验证出边界情况(如$n = 1$)的最优解 ,然后再证明：对于每个$F(n)$, 都可以由$F(n)$推导出结果$F(n + 1)$

#1103 旅行者的背包

题目描述

一个旅行者有一个最多能装 M 公斤的背包，现在有 n 种物品，它们的重量分别是 w_i ，它们的价值分别是 v_i 元/公斤，求旅行者能获得最大总价值。

尽可能的选择单位价值大的装入背包

输入

第 1 行：两个整数，M 背包容量（$M \leq 1000$）和 N 物品数量（$N \leq 30$）；
第 2 至 $N+1$ 行：每行两个整数 w_i, v_i ,表示每个物品的重量和价值。

输出

一个数，表示最大总价值。

样例输入

```
100 3
40 2
50 3
70 3
```

样例输出

```
300
```

#375 最大整数

题面描述

设有 n （$n \le 1000$ ）个正整数（每个在 $long\ long$ 范围内），将它们连接成一排，组成一个最大的多位整数。

输入第一行 1 个整数 n

第二行为 n 个正整数，分别用空格分隔输出一行，一个数，表示连接成的最大整数。

输入样例1

```
3
13 312 343
```

输出样例1

```
34331213
```

样例输入2

```
4
7 13 4 246
```

样例输出2

```
7424613
```

要让数字最大,高位需要尽可能大

使用$string$存储各个数值

按照字典序排序即可?

52 523 5

#614、数列分段

题目描述

对于给定的一个长度为 N 的 **正整数** 数列 A_i，现要将其分成 **连续的** 若干段，并且 **每段和** 不超过 M（可以等于 M），问最少能将其分成多少段使得满足要求。

输入格式

第一行包含两个正整数 N, M，表示了数列 A_i 的长度与每段和的最大值；
第二行包含 N 个空格隔开的非负整数 A_i。

输出格式

输出文件仅包含一个正整数，输出最少划分的段数。

样例输入

```
5 6
4 2 4 5 1
```

样例输出

```
3
```

数据范围与提示

对于 20% 的数据，有 $N \le 10$；
对于 40% 的数据，有 $N \le 1000$；
对于 100% 的数据，有 $N \le 10^5, M \le 10^9$，M 大于所有数的最大值，A_i 之和不超过 10^9。

尽可能累加,当不能累加时分段

#996 删数问题

题目描述

键盘输入一个高精度的正整数 $n(n \leq 10^{250})$ ，

去掉任意 s 个数字后剩下的数字按原左右次序将组成一个新的正整数。

编程对给定的 n 和 s ，寻找一种方案，使得剩下的数最小。

输入

第一行，一个不超过 250 位的整数。
第二行，一个整数 s 。

输出

删除 s 个整数后，保持原顺序的最小整数，前缀 " 0 " 不输出。

样例输入

```
178543
4
```

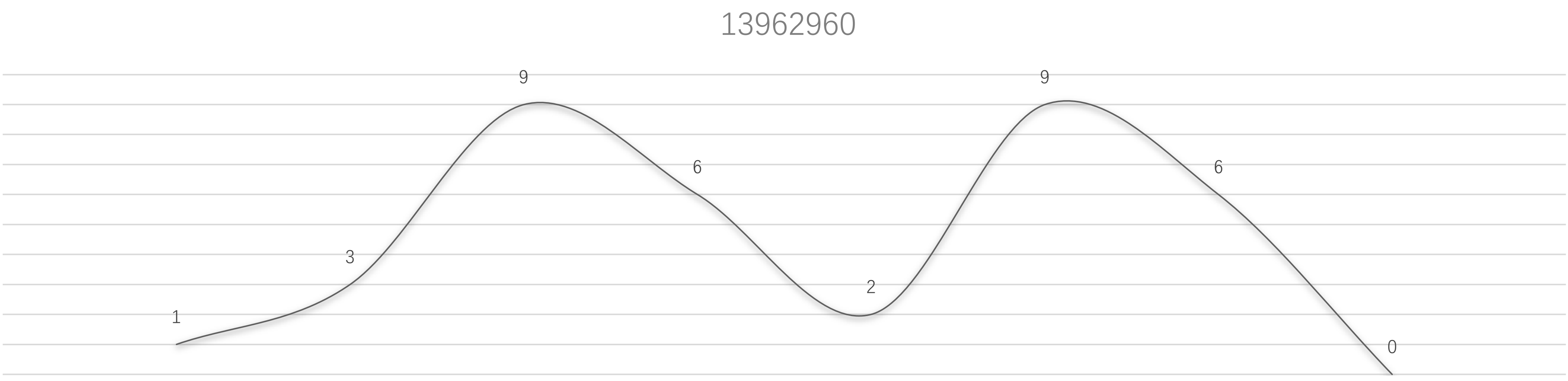
样例输出

```
13
```

删除最大的s个数位?

1003?

#996 删数问题



若数字序列是非递减的 ，如：1344,123那么只需要删除最后的S位数字

若数字序列是有增有减的,如：17283,434434那么找到第一个下标i,满足i上的数字大于$i+1$上的数字,删除i上的数字

对于无意义的前导0应当删除(如1001)

#999 排队接水

问题描述

有 n 个人在一个水龙头前排队接水，假如每个人接水的时间为 T_i，请编程找出这 n 个人排队的一种顺序，使得 n 个人的平均等待时间最小。

输入格式

输入，第一行为 n；
第二行分别表示第 1 个人到第 n 个人每人的接水时间 $T_1, T_2, \ldots, T_n$，每个数据之间有 1 个空格。

输出格式

输出两行，第一行为一种排队顺序，即 1 到 n 的一种排列；(如果多种排队方式，请保证先来的同学排在前面)
第二行为这种排列方案下的平均等待时间(输出结果精确到小数点后两位)。

输入样例

```
10
56 12 1 99 1000 234 33 55 99 812
```

输出样例

```
3 2 7 8 1 4 9 6 10 5
291.90
```

数据规模

对于全部的数据 $n \leq 1000, t \leq 10^6$

设一个等待时间序列为$T_1, T_2, \cdots, T_n$

不难发现总等待时间为

$$\sum_{i=1}^{n} T_i \times (n-i)$$

要使平均等待时间最小(总等待时间最小), 让序列单调非降即可

对序列升序排序即可

注意需要使用稳定的排序算法

#908 合并果子

题目描述

在一个果园里，多多已经将所有的果子打了下来，而且按果子的不同种类分成了不同的堆。多多决定把所有的果子合成一堆。

每一次合并，多多可以把两堆果子合并到一起，消耗的体力等于两堆果子的重量之和。
可以看出，所有的果子经过 $n-1$ 次合并之后，就只剩下一堆了。多多在合并果子时总共消耗的体力等于每次合并所耗体力之和。

因为还要花大力气把这些果子搬回家，所以多多在合并果子时要尽可能地节省体力。假定每个果子重量都为 1，并且已知果子的种类数和每种果子的数目，你的任务是设计出合并的次序方案，使多多耗费的体力最少，并输出这个最小的体力耗费值。

例如有 3 种果子，数目依次为 1，2，9。可以先将 1、2 堆合并，新堆数目为 3，耗费体力为 3。接着，将新堆与原先的第三堆合并，又得到新的堆，数目为 12，耗费体力为 12。所以多多总共耗费体力 $3+12=15$。可以证明 15 为最小的体力耗费值。

输入

包括两行，第一行是一个整数 n（$1 \leq n \leq 30000$），表示果子的种类数。第二行包含 n 个整数，用空格分隔，第 i 个整数 a_i（$1 \leq a_i \leq 20000$）是第 i 种果子的数目。

输出

包括一行，这一行只包含一个整数，也就是最小的体力耗费值。输入数据保证这个值小于 2^{31}。

样例输入

```
3
1 2 9
```

样例输出

```
15
```

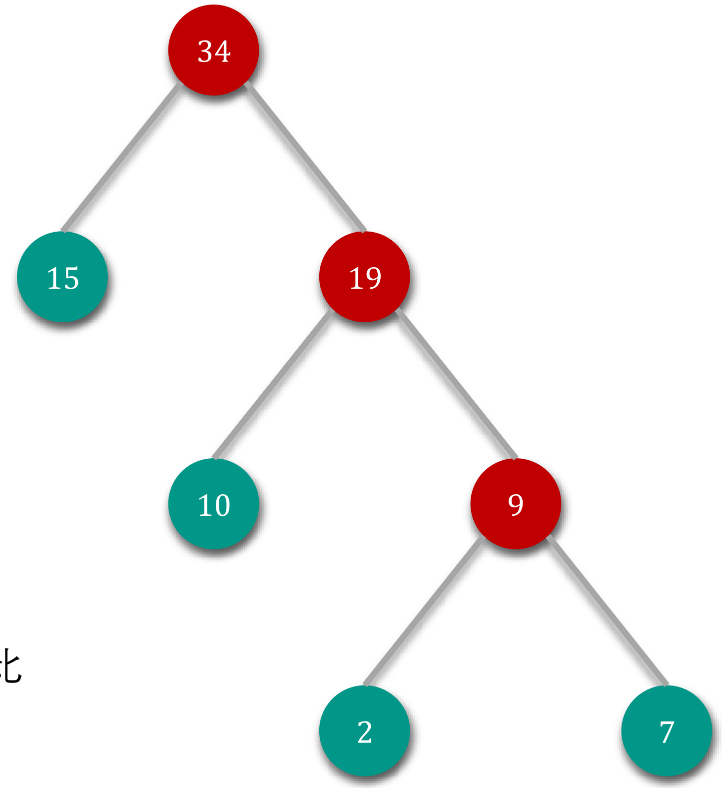
【数据规模】

对于 30% 的数据，保证有 $n \leq 100$；
对于 50% 的数据，保证有 $n \leq 5000$；
对于全部的数据，保证有 $n \leq 30000$。

#908 合并果子

不难发现对于任意 n 堆合并的过程都是类似与右图的树形结构

其中绿色节点为果子重量,红色节点为合并代价,线段代表一次搬运过程



总代价可看作各叶节点权值乘上它到根的距离

对于$n = 1$ 或 $n = 2$ 时结论显然成立

假设 $n = k$ 时该算法正确,考虑证明 $n = k + 1$ 时该算法也正确

可以发现：对于任意最优解,最小数一定在最深的那一层(否则可以将最小数与最深的那一层的某个比它大的数交换,这样得到的总代价是更小的),显然在最深的层的节点先合并的

综上 $n = k + 1$ 时第一步一定是合并最小的两个点

然后就转化到 $n = k$ 时场景,由归纳法,命题得证

#908 合并果子

对于$n=1$或$n=2$结论显然成立

设对于n堆,选取两堆重量合并代价最小且当前合并总代价为C

对于$n-1$堆,记任选两堆合并的代价为ΔC,要使得当前$C+\Delta C$最小,那么ΔC为最小两堆重量之和

```
priority_queue<int, vector<int>, greater<int>> q;
for (int i = 0; i < n; i++)
  q.push(a[i]);
while (q.size() > 1)
{
  int a = q.top();
  q.pop();
  int b = q.top();
  q.pop();
  ans += a + b;
  q.push(a + b);
}
```

对于每次取出两个最小值,加入一个新的值,需要维护动态有序性

若每一轮重新排序时间复杂度$O(n\log n)$,总时间复杂度$O(n^2\log n)$

可以使用 $priority\_queue$(堆)进行维护,总时间复杂度$O(n\log n)$

该模型可解决Huffman编码问题

#1691、牛奶供应

实验舱 青少年编程 走近科学 走进名校

题目描述

有一家牧场每天都会产出牛奶,在第 i 天,牛奶的产量为 p_i

批发商每天都会上门来收购,在第 i 天,收购量为 c_i , 如果某天收购量不大,多余的牛奶就会放进冷库保存

牛奶有一个保鲜期,如果超过了 m 天 （m 为一个给定的整数）,就必须倒掉了
卖给批发商时,应该先卖冷藏时间长的牛奶

给定天数 n 以及每天的牛奶产量和收购量,请求出牧场一共可以卖出多少量的牛奶

注意若某天的收购量很大,超过了当时可卖的总量,则当天卖出的牛奶数量就是可卖的总量

输入格式

第一行：两个整数 n 和 m

第二行到第 $n+1$ 行：第 $i+1$ 行每行两个整数表示 p_i 和 c_i

输出格式

单个整数表示答案

数据范围

对于 30% 的数据, $1 \le n, m \le 10000$
对于 100% 的数据, $1 \le n, m \le 100000, 0 \le p_i, c_i \le 10000$

输入样例1

```
5 2
50 0
100 0
250 0
300 0
1000 5000
```

输出样例1

```
1550
```

样例解释1

最后一天的收购量很大,但第一天和第二天的牛奶由于过期不能出售了

#998 均分纸牌

实验舱 青少年编程 LAB 走近科学 走进名校

【题目描述】

有 n 堆纸牌，编号分别为 1，2， $\ldots, n$ 。每堆上有若干张，但纸牌总数必为 n 的倍数。可以在任一堆上取若干张纸牌，然后移动。

移牌规则为：在编号为 1 的堆上取的纸牌，只能移到编号为 2 的堆上；在编号为 n 的堆上取的纸牌，只能移到编号为 $n-1$ 的堆上；其他堆上取的纸牌，可以移到相邻左边或右边的堆上。

现在要求找出一种移动方法，用最少的移动次数使每堆上纸牌数都一样多。

例如 $n=4$， 4 堆纸牌数分别为： ① 9 ② 8 ③ 17 ④ 6

移动 3 次可达到目的：

从 ③ 取 4 张牌放到④（9 8 13 10）->从③取3张牌放到 ②（9 11 10 10）-> 从②取1张牌放到①（10 10 10 10）。

【输入】

n （n 堆纸牌， $1 \leq n \leq 100$ ）

$a_1\ a_2\ \ldots\ a_n$ （n 堆纸牌，每堆纸牌初始数， $l \leq a_i \leq 10000$ ）。

【输出】

所有堆均达到相等时的最少移动次数。

【输入样例】

```
4
9 8 17 6
```

【输出样例】

```
3
```

#998 均分纸牌

实验舱 青少年编程 走近科学 走进名校 LAB

注意到每对相邻的纸牌最多只会移动一次我们记 x_i 为 i 到 $i-1$ 移动的纸牌数量

注意 x_i 有 $2 \le i \le n$,且 x_i 可以是负数,表示反方向移动

设平均数为 $\bar{a}$

x_i 的计算方式如下:

$$a_1 + x_2 = \bar{a}$$

$$a_2 - x_2 + x_3 = \bar{a} \Leftrightarrow x_2 = a_2 + x_3 - \bar{a}$$

$$a_{n-1} + x_n - x_{n-1} = \bar{a} \Leftrightarrow x_{n-1} = a_{n-1} + x_n - \bar{a}$$

$$\cdots$$

$$a_n - x_n = \bar{a} \Leftrightarrow x_n = a_n - \bar{a}$$

从左到右考虑 i若 x_i 不为 0,说明 a_i 到 a_{i+1}至少存在一次转移,最少移动次数为

$$\sum_{i=2}^{n}(x_i \neq 0)$$

```cpp
#include <bits/stdc++.h>
using namespace std;
int n, ans, a[101], ave;
int main()
{
  cin >> n;
  for (int i = 0; i < n; i++)
    cin >> a[i], ave += a[i];
  ave /= n;
  for (int i = 0; i < n; i++)
  {
    ans += (a[i] != ave);
    a[i + 1] -= ave - a[i];
  }
  cout << ans;
  return 0;
}
```

实验舱
青少年编程
走近科学 走进名校

谢谢观看