

7.7 省队训练模拟赛

rqgao2014

July 6, 2018

题目名称	冒泡排序	二分查找	城市规划
输入文件	bubble.in	binary.in	road1~6.in
输出文件	bubble.out	binary.out	road1~6.out
时间限制	2s	4s	-
内存限制	512MB	512MB	-
下发文件	无	有	有
题目类型	传统	传统	提交答案
有无 spj	无	无	有
编译开关	-O2 -std=c++11	-O2 -std=c++11	-

注意：评测时的栈空间大小不做单独限制，但使用的总空间大小不能超过内存限制。

Problem A. 冒泡排序 (bubble.c/cpp/pas)

Input file: bubble.in
 Output file: bubble.out
 Time limit: 2 seconds
 Memory limit: 512 megabytes

众所周知，冒泡排序是指以下这个伪代码。

算法 1 冒泡排序

输入: b_1, \dots, b_n

```

1: for  $i = 1 \rightarrow n - 1$  do
2:   for  $j = 1 \rightarrow n - i$  do
3:     if  $b_i > b_j$  then
4:       swap( $b_i, b_j$ )
5:     end if
6:   end for
7: end for
  
```

经过一定的测试，小 Y 发现冒泡排序的瓶颈在于 *swap* 函数执行地过于慢，所以对于已知的 b 序列，小 Y 关心的只有 *swap* 函数的执行数。

由于 b 序列过于庞大，执行程序之前小 Y 已经把它压缩成了 a 序列。以下为解压缩序列的方法：

- 最初，指针指向压缩后序列的第一项，并将栈清空。（这个栈中的每一个元素是一个序列。）最终栈中留下的唯一一个序列即为解压缩后的序列。
- 若指针对应的值 a_i 为正数，将 a_i 作为一个序列压入栈顶，并将指针定位为下一项。
- 若指针对应的值 a_i 为 0，将栈顶序列与弹栈后栈顶序列从栈中取出，将前者序列放在后者序列的尾端相接，将新序列压入栈顶，并将指针定位为下一项。
- 若指针对应的值 a_i 为负数，取出栈顶的序列，然后将 $|a_i|$ 个该序列首尾相接组成的行序列压入栈顶，并将指针定位为下一项。

Input

第一行包含一个整数 n 。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ，即压缩后的序列。

Output

唯一一行包含一个整数，即 *swap* 数模 $10^9 + 7$ 。

Examples

bubble.in	bubble.out
9 1 2 3 0 -4 5 0 0 -2	45

Notes

设数据中有 L 个 i 满足 $a_i \neq 0$ 。

对于 100% 的数据： $1 \leq n$; $|a_i|, L \leq 3 \cdot 10^5$; 保证最后栈中有且仅有一个序列。

子任务 1[20 分]： $a_i \geq 0$ 。

子任务 2[20 分]： $L \leq 3000$ 。

子任务 3[20 分]： $L \leq 10^5, a_i \leq 5000$ ，小于 0 的 a_i 数量不超过 5000。

子任务 4[20 分]： $|a_i|, L \leq 10^5$ 。

子任务 5[20 分]：无特殊限制。

Problem B. 二分查找 (binary.c/cpp/pas)

Input file: `binary.in`
Output file: `binary.out`
Time limit: 1 seconds
Memory limit: 512 megabytes

同样众所周知，二分查找是指以下这个伪代码。

算法 2 二分查找

输入: b_1, \dots, b_n 和其他若干信息

输出: R

```
1: function FindMid( $L, R$ )
2:   return 一个  $[L, R - 1]$  之间的正整数
3: end function
4:
5: function Check( $Mid$ )
6:    $Cnt = Cnt + b_{Mid}$ 
7:   ...
8:   return 0/1
9: end function
10:
11:  $L = 1, R = n + 1$ 
12: while  $L < R$  do
13:    $Mid = FindMid(L, R)$ 
14:   if Check( $Mid$ ) then
15:      $R = Mid$ 
16:   else
17:      $L = Mid + 1$ 
18:   end if
19: end while
```

给定 b_1, b_2, \dots, b_n 。试求在 *FindMid* 的最优决策下，最坏的情况下 Cnt (Cnt 的初值为 0) 为多少。

Input

第一行一个整数 n 。

第二行一个只含有 '1' ~ '9' 的长度为 n 的字符串 s , b_i 的数值即为 s 中的第 i 个字符对应的数字。

Output

一行共一个整数，表示最坏情况下 Cnt 的最小值。

Examples

binary.in	binary.out
3 111	2
4 1111	3
6 111119	10
详见下发文件 <code>ex_binary4.in</code>	详见下发文件 <code>ex_binary4.ans</code>

Notes

对于所有数据，满足 $n \leq 10^6$ 。

子任务 1[12 分]： $n \leq 500$

子任务 2[23 分]： $n \leq 5000$

子任务 3[34 分]： $n \leq 3 \cdot 10^5$

子任务 4[31 分]：无特殊限制

Problem C. 道路规划 (road1~6.out)

Input file: road*.in
Output file: road*.out
Time limit: -
Memory limit: -

已知一个国家内有 n 座城市和 $n - 1$ 条双向道路连接，且两两城市间可以相互到达，现在希望再建设 K 条道路，使得两两城市间的最短路的和尽可能小。

Input

第一行三个整数 n, K, w_0 ，表示点数、新建道路数和评分参数。

接下来 $n - 1$ 行，每行两个整数 u, v 表示初始的道路。

Output

输出共 K 行，每行两个整数，表示新建的边。

Examples

road*.in	road*.out
4 1 8 1 2 2 3 3 4	1 4

Notes

对于所有数据，满足 $n, K \leq 1000$ 。

评分方式：如果输出不合法，得分为 0 分。否则，设 w 为加入输出中 K 条边后两两点最短路的和。如果 $w \leq w_0$ ，那么得分为 10 分；否则得分为 $\lfloor P \times 50^{1.0-w/w_0} \rfloor$ 分。

分值分布：对于测试点 1， $P = 10$ ；对于其余测试点， $P = 18$ 。