CSP2022模拟题1020

8:00--12:00

测试结束后，监考老师开网后，在CWOI上提交，需要文件操作

考后稍事休息，保持安静，12:23分才能放学

A. 初赛

qksort.cpp/.in/.out 1S 512M

题目描述

初赛经常考排序算法，于是婷姐给大家准备了一道。

婷姐写了一个快排：

```
// sort a[0], a[1], ..., a[n - 1]
void qsort(int a[], int n) {
  // ++cnt;
  if (n <= 1) return;
  int mid = 1;
  for(int i = 1; i < n; i++) {
    // ++cnt;
    if (a[i] < a[0]) {
      swap(a[i], a[mid++]);
    }
  }
  swap(a[0], a[--mid]);
  qsort(a, mid);
  qsort(a + mid + 1, n - mid - 1);
}
```

显然这份代码是有破绽的，而婷姐希望你能够找出它的破绽。具体地说你需要构造一个序列 a 使得上述算法的复杂度达到 $O(n^2)$。

不过只是这样会有点简单。

考虑到 CSP-J 组的分数线比 CSP-S 组的要高，婷姐也给 CSP-J 组的同学们上了一点难度。她稍微魔改了一下代码：

```
int randint() {
    static unsigned s = 0x80000001;
    unsigned long t = s;
    t ^= t >> 10;
    t ^= t << 9;
    t ^= t >> 25;
```

```
    s = t;
    return (t & INT_MAX);
}

// sort a[0], a[1], ..., a[n - 1]
void qsort2(int a[], int n) {
  // ++cnt;
  if (n <= 1) return;
  swap(a[0], a[randint() % n]);

  int mid = 1;
  for (int i = 1; i < n; i++) {
    // ++cnt;
    if (a[i] < a[0]) {
      swap(a[i], a[mid++]);
    }
  }
  swap(a[0], a[--mid]);

  int mid2 = mid + 1;
  for (int i = mid + 1; i < n; i++) {
    // ++cnt;
    if (a[i] <= a[mid]) {
      swap(a[i], a[mid2++]);
    }
  }
  swap(a[mid], a[--mid2]);

  qsort2(a, mid);
  qsort2(a + mid2 + 1, n - mid2 - 1);
}
```

同样地，你需要构造一个序列 a 使得上述算法的复杂度达到 $O(n^2)$。

在本题中，快排算法的实际运行复杂度由累加器变量 `cnt` 表征（见代码中注释掉的 `++cnt` 语句），因此你可以理解为本题实际上要求你构造序列 a 使得最终 `cnt` 的值大于某个规定阈值。具体的要求见**读入格式**和**数据范围**。

读入

一行三个非负整数 t, m, d。

t（$0 \le t \le 1$）表示你需要 hack 的快排算法的种类。0 表示 `qsort`，1 表示 `qsort2`。

m 表示你构造的序列的长度的最大值。

d 表示你需要让 `cnt` 最终的值大于的阈值。

输出

你需要输出 2 行。

第一行一个整数 n（$n \leq m$），接下来一行 n_1 个整数 $a_1, a_2, \ldots, a_n$（$1 \leq a_i \leq 10^9$）。

数据范围

本题保证，$t = 0$ 和 $t = 1$ 的测试点各占一半。

对于所有数据 $m \leq 500, d \leq 10^5$。

B. 序列

sequence.cpp/.in/.out 1S 512M

题目描述

婷姐喜欢手玩序列。

对于一个 n 元整数序列 $a_1, a_2, \ldots, a_n$，她会构造一个序列 $b_1, b_2, \ldots, b_n$。其中 b_i 要么是 a_i 的前缀最小值，即 $\min(a_1, a_2, \ldots, a_i)$，要么是 a_i 的后缀最小值，即 $\min(a_i, a_{i+1}, \ldots, a_n)$。至于究竟是哪个取决于她当时构造 b_i 时的心情。

现在婷姐已经记不得序列 a，也记不得她在手玩序列时曾经涌现过的种种复杂心情，只记得了序列 $b_1, b_2, \ldots, b_n$。她希望你帮她还原出任意一个可能的 $a_1, a_2, \ldots, a_n$。

可惜现实是残酷的，如果她一不小心记错了，导致无法还原出合法的 $a_1, a_2, \ldots, a_n$，也请你不要告诉她，免得让她伤心，不妨给她讲点开心的事。也就是说这时你需要输出 `cut_the_watermelon_with_a_hand`。

读入

第一行一个正整数 n。

第二行 n 个正整数 $b_1, b_2, \ldots, b_n$。

输出

若有解，输出任意一个，一行 n 个数表示 $a_1, a_2, \ldots, a_n$。

否则输出 `cut_the_watermelon_with_a_hand`。

数据范围

对于 10% 的数据 $n \leq 10$。

对于另外 30% 的数据 $b_i \leq 2$。

对于所有数据 $n \leq 1 \times 10^5, 1 \leq b_i \leq 10^5$。

C. 匹配

match.cpp/.in/.out 1S 512M

题目描述

婷姐造了一棵 n 个点的树（n 为偶数），她希望将树上的点两两匹配。也就是说对于任意一个结点 u，**有且仅有**一个结点 v 与之配对。

不过婷姐对于匹配有一些限制。对于两个匹配 (u,v) 和 (x,y)，婷姐要求这两个匹配要么相离，要么一个包含另一个。

- 相离：u 到 v 的路径与 x 到 y 的路径不相交。
- 包含：u 到 v 的路径包含 x 到 y 的路径或者 x 到 y 的路径包含 u 到 v 的路径。

婷姐需要你告诉他，这样的匹配一共有多少种方案。答案对 998244353 取模。

两个方案不同当且仅当存在某个结点，使得在这两个方案中与它配对的那个结点不同。

读入

第一行一个整数 n。

接下来 $n-1$ 行每行 2 个整数 x,y，表示树上的一条边。

输出

一行一个整数表示方案数，对 998244353 取模。

数据范围

保证 n 是偶数。

对于 20% 的数据 $n \leq 8$。

对于 50% 的数据 $n \leq 100$。

对于所有数据 $2 \leq n \leq 2000$。

D. T语言

t.cpp/.in/.out 1S 512M

题目描述

虽然你们大概率不知道婷姐是谁，但是婷姐设计的 T 语言你们是必须要知道的。

每个 T 语言表达式在一行中出现，每个表达式都有一个值。

需要注意的是，T 语言中的运算符**没有像 C/C++ 中的优先级**，而是**从右到左**进行运算，也就是说乘法和加法的优先级是一样的。当然括号可以改变运算的顺序。

下面给出 T 语言表达式的一些实例。

T 语言表达式	解释
`var = 1 2 3`	将整数序列 $[1,2,3]$ 存储在var中，表达式的值是 `1 2 3`
`var + 4`	var 中 每个元素的值增加 4，整个表达式的值为 `5 6 7`; 注意： var 存储的内容没有改变的，仍然为 `1 2 3`
`var - 4`	var 中 每个元素的值减少 4，整个表达式的值为 `-3 -2 -1`; 注意： var 存储的内容没有改变的，仍然为 `1 2 3`
`var * 4`	var 中 每个元素的值乘上 4，整个表达式的值为 `4 8 12`; 注意： var 存储的内容没有改变的，仍然为 `1 2 3`
`- / var`	`-` 运算符被插在 var 中的每两个元素之间（即：`1-2-3`），**计算结果存储到 var**，表达式的值为 `2`。
`+ / var`	`+` 运算符被插在 var 中的每两个元素之间（即：`1+2+3`），**计算结果存储到 var**，表达式的值为 `6`。
`* / var`	`*` 运算符被插在 var 中的每两个元素之间（即：`1*2*3`），**计算结果存储到 var**，表达式的值为 `6`。
`a + ( a = 5 ) + a + ( a = 6 )`	表达式的值为 22

现在想请你编写一个 T 语言的解释器。计算给出的每一个 T 语言表达式的值。

输入的数据均为正整数，且小于 10^5；所有计算过程中的任意整数值（中间结果）的绝对值保证在 `long long` 范围内。

提供的算术运算符只有 `+`, `-`, `*` 。

变量名由 1 到 3 个小写字母组成，用一个空格分隔元素（常数、变量、括号，运算符）。

如果存在并行的（无法决定顺序的）子序列计算，默认从右往左计算。

输入

第一行输入一个正整数 N（$1 \leq N \leq 10$）。

第 2 行 ~ 第 $N+1$ 行， 每行输入一个 T 语言表达式。每个表达式不超过 150 个字符，表达式中可能包括空格符。

题目保证给出的数据中，没有涉及到序列与序列的加法和乘法。

输出

对于输入中的每一行，在一行中输出该表达式的值。

样例

输入1

```
4
var = 1 2 3
var + 4
- / var
a + ( a = 5 ) + a + ( a = 6 )
```

输出1

```
1 2 3
5 6 7
2
22
```

输入2

```
3
( 3 - 2 ) - 4
var = 1 2 3
var * ( a = 2 )
```

输出2

```
-3
1 2 3
2 4 6
```

数据范围约束

对于其中 10% 的数据保证，不会出现任何变量，且任意表达式不会含有括号；

对于另外 10% 的数据保证，不会出现任何变量；

对于另外 20% 的数据保证，变量的内容均为单个数字，且任意表达式不会含有括号；

对于另外 20% 的数据保证，变量的内容均为单个数字；

对于另外 20% 的数据保证，任意表达式不会含有括号；

对于另外 20% 的数据保证，没有任何特殊限制。