

# 2019 年非专业级软件能力认证

CCF-CSP-2019

## 提高级（第二轮）第一次认证

认证时间：2019 年 11 月 16 日 8:30~12:00

题目名称	近海之主	养马	膜拜大会
题目类型	传统型	传统型	传统型
目录	treasure	horse	fake
可执行文件	treasure	horse	fake
输入文件名	treasure.in	horse.in	fake.in
输出文件名	treasure.out	horse.out	fake.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒
内存限制	512MB	512MB	512MB
测试点数目	10	20	10
每个测试点分值	10	5	10

### 提交源程序文件名

对于 C++ 语言	treasure.cpp	horse.cpp	fake.cpp
对于 C 语言	treasure.c	horse.c	fake.c

### 编译选项

对于 C++ 语言	-lm	-lm	-lm
对于 C 语言	-lm	-lm	-lm

### 注意事项：

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. 除非特殊说明，结果比较方式均为忽略行末空格及文末回车的全文比较。
3. C/C++中的函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
4. 全国统一评测时采用的机器配置为：Intel(R) Core(TM) i3-4170 CPU @ 3.70GHz，内存 4G，上述时限以此配置为准。
5. 只提供 Linux 格式附加样例文件。
6. 评测在 Ubuntu 18.04 下进行。

# 近海之主

(treasure.cpp/c)

## 【问题描述】

鲳鱼，又名近海之主。最近，他在他的近海领地内发现了一个远古遗迹，这个远古遗迹有 $n$ 个房间，每个房间里放了一个宝藏。不幸的是，每个房间必须有它的钥匙才能进入，而这些房间的钥匙要么被放在了其他的房间里，要么在邪恶的KLL手上。

刚开始，鲳鱼从KLL手上勒索了某些房间的钥匙(也可能一个都勒索不到)，他现在要按随机的顺序访问所有房间，每次他会随机访问一个他**没有访问过**的房间，如果有这个房间的钥匙，那么他就可以进入，并取得这个房间里的宝藏和钥匙。

注意，如果一个房间被试图访问但没有成功进入，那么鲳鱼就不会再访问这个房间了。

现在，鲳鱼想要知道期望意义下，他能拿到多少个宝藏。

## 【输入格式】

输入文件名为treasure.in。

第一行一个整数 $n$ ,表示宝藏的个数。

接下来一行 $n$ 个整数，第 $i$ 个整数表示第 $i$ 个房间的钥匙的所在地 $a_i$ ，如果第 $a_i = -1$ ，表示第 $i$ 个房间的钥匙一开始就被鲳鱼获得了。

## 【输出格式】

输出文件名为treasure.out。

输出一行，表示能得到宝藏的期望个数。

可以证明期望一定是个分数 $\frac{x}{y}$  ( $\gcd(x, y) = 1$ )的形式，你只需要求出这个分数模998244353的结果：

$$x \times y^{-1} \bmod 998244353 = x \times y^{998244351} \bmod 998244353$$

## 【输入输出样例1】

treasure.in

treasure.out

5

166374062

-1 1 1 3 -1

## 【样例1解释】

鲳鱼一共有120种合法的访问顺序，其中能得到宝藏数为2, 3, 4, 5的情况数分别是40(如54321), 35(如54312), 30(如54132), 15(如12345)，所以答案是 $\frac{19}{6}$ 。

## 【输入输出样例2】

请选手见下发文件treasure2.in和treasure2.ans。

## 【数据规模与约定】

测试点编号	n=	其他特殊性质	
1	10	保证存在一种得到所有宝藏的顺序	无
2			
3	100		
4			
5	99998		$a_i=-1$
6	99999		若 $a_i \neq -1$ , 则 $a_{a_i}=-1$
7			
8	100000	无如上保证	无
9			
10			

对于100%的测试数据和样例，保证  $1 \leq n \leq 10^5, a_i \in [1, n] \cup \{-1\}$ 。

# 养马

(horse.cpp/c)

## 【问题描述】

鲑鱼养了一匹朝鲜马，这天鲑鱼打算带它出去散步。

于是鲑鱼来到了非法树下。众所周知，非法树上有 $n$ 个点，点与点之间由 $n - 1$ 条道路连接而成(保证所有的点是联通的)，而道路上存在障碍，朝鲜马需要跳过这个障碍，则需要花费一定的体力。同时朝鲜马也可以吃树或者休息，来恢复一定的体力。如果朝鲜马吃掉非法树的一个点，它可以恢复 $a_i$ 点体力。被吃掉的点也可以经过。朝鲜马也可以休息一个单位时间来恢复一点体力。

注意：任何时刻朝鲜马的体力值不能为负数，但是可以是0或无穷大。一个点只能吃一遍，但障碍每次跳过去都会消耗体力；朝鲜马跳过障碍或者吃树都不需要时间；由于树枝比较脆弱，朝鲜马只能经过每条边最多两次。

现在鲑鱼带着朝鲜马从树的根节点——一号点开始散步，初始时马的体力为0。他们要在遍历完所有点之后回到一号点。由于等待朝鲜马恢复体力是十分无聊的，所以他想问问你遍历所有点并回到原点所需的最小等待时间是多少。

## 【输入格式】

输入文件名为horse.in。

第一行一个整数 $n$ ，表示非法树上一共有 $n$ 个点。

接下来一行 $n$ 个整数，第 $i$ 个整数 $a_i$ 表示第 $i$ 个树的提供的体力值。

接下来 $n - 1$ 行，第 $i$ 行三个整数 $u_i, v_i, w_i$ ，表示树上的一条边。

## 【输出格式】

输出文件名为horse.out。

一行整数，表示朝鲜马最少需要休息的时间。

## 【输入输出样例1】

horse.in	horse.out
5	23
4 2 1 5 7	
1 2 4	
1 3 5	
4 2 9	
5 2 3	

## 【样例1解释】

可以按照 $1 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 2 \rightarrow 1$ 的顺序遍历。

## 【输入输出样例2】

请选手见下发文件horse2.in和horse2.ans。该样例符合11~14测试点的特征。

## 【数据规模与约定】

测试点编号	n=	特殊性质
1	11	无
2		
3	99994	所有树边边权=0
4	99995	只存在一条树边边权>0
5		
6	99996	第 i 条边连接 i 与 i+1
7		
8	99997	一个点最多 2 个儿子
9		
10		
11	1000	一个点最多 10 个儿子
12		
13		
14		
15	99999	第 i 条边连接 1 与 i+1
16		
17	100000	无
18		
19		
20		

对于100%的测试数据和样例，保证 $1 \leq n \leq 10^5, 1 \leq u_i, v_i \leq n, 0 \leq w_i, a_i \leq 10^9$ ，给出的是一棵树。

# 膜拜大会

(fake.cpp/c)

## 【问题描述】

最近，鲳鱼开始参加各种膜拜大会。

膜拜大会是一种由 $n$ 个人进行的大型活动，其中每个人都有编号，且鲳鱼的编号是1。这 $n$ 个人坐成一圈，其中 $i$ 号的左边是 $i - 1$ 号，右边是 $i + 1$ 号（例外地，1号的左边是 $n$ 号， $n$ 号的右边是1号）。不妨记 $i$ 号左边人的编号为 $L_i$ ，右边人的编号为 $R_i$ 。

在膜拜大会中，每一个人初始有一个巨佬值，第 $i$ 个人的巨佬值为 $A_i$ 。唯一可以改变巨佬值的操作是装弱。如果一个人进行了装弱，那么他左边及右边的人的巨佬值都会加上他的巨佬值，而他自己的巨佬值随后清零。简单地说，如果第 $i$ 个人装弱，相当于将 $A_{L_i}$ 以及 $A_{R_i}$ 加上 $A_i$ ，随后将 $A_i$ 变为0。

但是装弱不是随心所欲的。膜拜大会的规则规定，一场膜拜大会包含 $m$ 个装弱操作，而且每个人最多只能装弱一次。而且装弱的顺序不是参加者指定，而是在所有满足以上条件的顺序中随机的。即，总共 $m$ 次装弱操作，每次都随机一个没有装弱过的人进行装弱。

现在，给出每场膜拜大会中的 $n$ 和 $m$ ，以及对于每个 $i$ 初始的 $A_i$ ，鲳鱼想要知道膜拜大会结束后他自己的巨佬值（即 $A_1$ ）大于等于 $K$ 的概率。

## 【输入格式】

输入文件名为fake.in。

第一行一个整数 $T$ ，表示鲳鱼参加的大会场数。

接下来共 $T$ 组数据，每组包含两行，其中第一行包含三个整数 $n, m, K$ ，意义如上文所示；第二行包含 $n$ 个整数 $A_1, A_2, \dots, A_n$ ，表示每个人初始的巨佬值。

## 【输出格式】

输出文件名为fake.out。

输出文件共 $T$ 行，表示对于每组数据的概率。

可以证明概率一定是个分数 $\frac{x}{y}$  ( $\gcd(x, y) = 1$ )的形式，你只需要求出这个分数模998244353的结果：

$$x \times y^{-1} \bmod 998244353 = x \times y^{998244351} \bmod 998244353$$

## 【输入输出样例1】

fake.in

fake.out

1  
5 3 10  
5 1 2 2 1

865145106

## 【样例1解释】

可以算出总共有60种装弱顺序，其中只有4种是符合条件的。这4种顺序是1 2 5、1 5 2、3 4 5、4 3 2。

以1 2 5这种顺序为例，初始所有人的巨佬值为5 1 2 2 1；1号装弱后，巨佬值变为0 6 2 2 6；2号装弱后，巨佬值变为6 0 8 2 6；5号装弱后，巨佬值变为12 0 8 8 0。最后 $A_1 = 12 \geq 10$ ，满足条件。可以算出其他所有方案的结果。

因此，最终答案为 $\frac{4}{60} = \frac{1}{15}$ ，取模后的值等于865145106。

【输入输出样例2】

请选手见下发文件fake2.in和fake2.ans。

【输入输出样例3】

请选手见下发文件fake3.in和fake3.ans。该样例符合6~7测试点的特征。

【数据规模与约定】

测试点编号	n=	特殊性质
1	9	无
2		
3	99997	m=1
4	99998	最多只存在一个 i 满足 $A_i > 0$
5		
6	99999	$A_1 \geq K$
7		
8	100000	无
9		
10		

对于100%的测试数据和样例，保证 $1 \leq T \leq 5$   $1 \leq m \leq n - 2 < n \leq 10^5, 0 \leq A_i, K \leq 10^9$ 。