提高级费客霞模拟赛

时间：2026 年 * 月 * 日 07:45 ~11:45

题目名称	皮配	水晶球	排序	
题目类型	传统型	传统型	传统型	传统型
目录	match	magic	sort	move
可执行文件名	match	magic	sort	move
输入文件名	match.in	magic.in	sort.in	move.in
输出文件名	match.out	magic.out	sort.out	move.out
每个测试点时限	1.0 秒	2.0 秒	2.0 秒	8 秒
内存限制	512 MiB	512 MiB	1024 MiB	1024 MiB
测试点数目	10	10	10	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	match.cpp	magic.cpp	sort.cpp	move.cpp

编译选项

对于 C++ 语言	-O2 -std=gnu++1z

皮配 (match)

【题目描述】

Kaguya 是一个喜欢图论的女孩子。她最近在学习图论中的最大匹配相关问题，并已经对这类问题略有心得。今天，Kaguya 得到了一张简单无向图 G。她很快得到了这张图的最大匹配。但她已经对如此经典的问题产生了厌倦，于是稍微修改了一下匹配的定义。

一张简单无向图边集的一个子集被 Kaguya 称为一个「皮配」，当且仅当其中任意两条边都有至少一个共用端点。

Kaguya 想知道 G 的最大「皮配」大小。她当然早已求出了答案，不过还是想让你也算一算。

另外，请注意：Kaguya 认为简单无向图是指无重边无自环的无向图。

【输入格式】

从文件 $match.in$ 中读入数据。
本题有多组测试数据。
输入的第一行包含一个整数 t , 表示测试数据组数。
接下来依次输入每组测试数据，对于每组测试数据:
输入的第一行包含两个整数 nm , 分别表示 G的点数与边数。
m 行，第 $i(1 \leq i \leq m)$ 行两个整数 u_i v_i , 表示 G 中的第 i 条边。

【输出格式】

输出到文件 $match.out$ 中。
对于每组测试数据输出一行一个整数，表示 G 的最大「皮配」大小。

【样例 1 输入】

```
3
4 1
2 3
1 0
5 4
3 2
5 3
5 1
4 5
```

【样例 1 输出】

```
1
0
3
```

【样例 1 解释】

对于第三组数据：最大「皮配」为 $\{(5,3),(5,1),(4,5)\}$。

【样例 2】

见选手目录下的 $match/match2.in$ 与 $match/match2.ans$

【子任务】

对于所有测试数据保证: $t \geq 1$, $1 \leq n \leq 10^6$, $0 \leq m \leq 10^6$, $0 \leq \sum n$, $\sum m \leq 10^6$ 。

每个测试点的具体限制见下表:

测试点编号	∑n, ∑m ≤	特殊性质
1	10	无
2	20	无
3	10^3	m = n −1,$u_i = i$,$v_i = i + 1$
4	10^3	m = n −1,$u_i = 1$,$v_i = i + 1$
5	10^3	m = n,$u_i = i$,$v_i = (i \bmod n) + 1$
6	10^3	无
7	10^6	m = n −1,$u_i = i$,$v_i = i + 1$
8	10^6	m = n −1,$u_i = 1$,$v_i = i + 1$
9	10^6	m = n,$u_i = i$,$v_i = (i \bmod n) + 1$
10	10^6	无

水晶球 (magic)

【题目描述】

Kaguya 是一个会魔法的女孩子。
在一次旅行中，她偶然得到了 n 个水晶球，并将其编号为 $1 \ldots n$。这些水晶球可以储存魔法，初始时均为活跃态。Kaguya 可以按任意顺序将其变为稳定态。经过一些计算，她发现，当第 i 个水晶球由活跃态变为稳定态时，会向第 l_i 至第 r_i 个水晶球中的所有活跃态水晶球各注入 c_i 单位魔法。Kaguya 希望最大化最终所有水晶球中的魔法总量。她当然已经计算出了最优的变化顺序，不过她觉得这个问题不算平凡，希望你也求一下这个最大值。

【输入格式】

从文件 $magic.in$中读入数据。
输入的第一行包含一个整数 n , 表示水晶球的数目。
n 行，第 i 行输入三个整数 l_i r_i c_i , 表示第 i 个水晶球的参数。

【输出格式】

输出到文件 $magic.out$ 中。
输出一行包含一个整数，表示最终所有水晶球中的魔法总量的最大可能值。

【样例 1 输入】

```
5
1 2 1
2 2 3
2 4 7
3 5 4
3 5 5
```

【样例 1 输出】

```
25
```

【样例 1 解释】

最优变化顺序为：5 →3 →1 →2 →4。

【样例 2】

见选手目录下的$magic/magic2.in$与 $magic/magic2.ans$

【子任务】

对于所有测试数据保证: $1 \leq n \leq 10^3$, $1 \leq l_i \leq i \leq r_i \leq n$, $max\{i-l_i, r_i-i\} \leq 7$, $0 \leq c_i \leq 10^9$

每个测试点的具体限制见下表:

测试点编号	$n \leq$	特殊性质
1	10	无
2	20	无
3	30	无
4	10^2	$max\{i-l_i, r_i-i\} \leq 0$
5	10^2	$max\{i-l_i, r_i-i\} \leq 1$
6	10^2	$max\{i-l_i, r_i-i\} \leq 2$
7	10^2	$l_i = max\{i-6, 1\}, r_i = min\{i+6, n\}, c_i = 1$
8	10^2	$l_i = max\{i-7, 1\}, r_i = min\{i+7, n\}, c_i = 1$
9	10^2	无
10	10^3	无

排序 (sort)

【题目描述】

Kaguya 是一个喜欢秩序的女孩子。她常常收到很多序列作为礼物，有时她希望对某些序列进行排序。今天，Ayu 又送了 Kaguya 一个排列 $a_{1...n}$ 。Kaguya 希望按照如下伪代码对其进行 排序:

```
function sort(a[1...len(a)]):
  if len(a) <= 1 then return a
  let pivot = a[ceil(len(a) / 2)]
  let al, ag = empty sequence
  for i in 1...len(a) do
    let cmpcnt = cmpcnt + 1
    if a[i] < pivot then append a[i] to al
    if a[i] > pivot then append a[i] to ag
  return sort(al) + [pivot] + sort(ag)
```

Kaguya 比较关心排序需要的比较次数，所以希望你求出：用该函数对 $a_{1...n}$ 进行一次排序后，$cmpcnt$会增加多少。

【输入格式】

从文件$sort.in$中读入数据。
输入的第一行包含一个整数 n, 表示要排序的排列长度。
输入的第二行包含 n 个整数 $a_{1...n}$, 表示要排序的排列。

【输出格式】

输出到文件$sort.out$中。
输出一行包含一个整数，表示一次排序后 $cmpcnt$ 的变化量。

【样例 1 输入】

```
5
4 3 5 1 2
```

【样例 1 输出】

```
11
```

【样例 1 解释】

作为函数参数的非空序列共$[4,3,5,1,2]$、$[4,3,1,2]$、$[4]$、$[1,2]$、$[2]$ 五个。

【样例 2】

见选手目录下的 $sort/sort2.in$与 $sort/sort2.ans$

【样例 3】

见选手目录下的 $sort/sort3.in$与 $sort/sort3.ans$

【样例 4】

见选手目录下的 $sort/sort4.in$与 $sort/sort4.ans$

【样例 5】

见选手目录下的 $sort/sort5.in$ 与 $sort/sort5.ans$

【子任务】

对于所有测试数据保证: $1 \leq n \leq 7 \times 10^5$, $1 \leq a_i \leq n$, $\forall i \neq j, a_i \neq a_j$

每个测试点的具体限制见下表:

测试点编号	n ≤	特殊性质
1	3,000	无
2	2×10^5	$a_{1 \ldots n}$随机生成
3	2×10^5	$a_i = i$
4	2×10^5	$\sum_i [a_i \neq i] \leq 10$
5	2×10^5	$\lvert a_i - i \rvert \leq 10$
6 ~ 8	10^5	无
9 , 10	7×10^5	无

想要成为影之实力者（move）

题目描述

暗影大人最近学会了新的能力：瞬移。
于是，他来到一棵树上进行实验。
但他对能力掌握得不够熟练。

具体地，他会从 1 号点开始，在 n 号点停止，瞬移任意次。
假设他当前所在的节点为 x，则他可以选择 $[x+1,n]$ 的任意一个点瞬移过去。
x 和他瞬移到的点在树上的距离就是这次瞬移的距离。

请你求出暗影大人瞬移的最大总距离。

输入格式

从文件`move.in`中读入数据。
第一行一个整数 n。
后面 $n-1$ 行，每行三个整数 x , y ,z，表示存在一条连接 x 和 y 的长度为 z 的边。

输出格式

输出到文件`move.out`中。
一行一个整数，表示最大的总距离。

样例输入 1

```
2
1 2 1
```

样例输出 1

```
1
```

样例输入 2

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

样例输出 2

```
53
```

样例输入 3

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

样例输出 3

```
-2
```

样例 3 解释

先从 1 瞬移到 3，距离为 1。然后从 3 瞬移到 4，距离为−3。总距离为−2。

【样例 4】

见选手目录下的 $move/move4.in$ 与 $move/move4.ans$

【样例 5】

见选手目录下的 $move/move5.in$ 与 $move/move5.ans$

测试点约束

每个测试点的具体限制见下表：

对于 100% 的数据，$1 \leq n \leq 10^6$，$|z| \leq 10^6$

子任务编号	n	特殊性质	分值
1	≤ 100	无	5
2	≤ 5000	无	10
3	$\leq 2 \times 10^4$	无	5
4	$\leq 10^5$	A	10
5	$\leq 10^5$	B	10
6	$\leq 10^5$	C	3
7	$\leq 10^5$	无	25
8	$\leq 3 \times 10^5$	无	16
9	$\leq 10^6$	无	16

说明：

A: 保证树的形态是一条链。

B: 保证 1 号点的度数为 n-1。

C: 保证 $z \geq 0$