

2019Noip模拟赛day1

题目概述

题目名称	Candies	Game	Tree
源文件名	<code>candies.c/cpp/pas</code>	<code>game.c/cpp/pas</code>	<code>tree.c/cpp/pas</code>
输入输出文件名	<code>candies.in/out</code>	<code>game.in/out</code>	<code>tree.in/out</code>
时间限制	1 sec	1 sec	2 sec
空间限制	128 MB	128 MB	256 MB
题目类型	传统	Special Judge	传统
是否开启 O2 优化	是	是	是

A. Candies(candies)

题目描述

小 W 有 N 堆糖果，每堆有 a_i 个。现在，他要把这 N 堆糖果分别分给 N 个小朋友。为了使每个小朋友被分到的糖果数尽量均匀，小 W 每次会按顺序执行以下操作：

1. 找到糖果数量最多的糖果堆（如果多于一个，则随机选择一个），从中取出一颗糖果；
2. 在剩下各堆糖果中(包含前面取出糖果的那堆)找到糖果数量最少的糖果堆（如果多于一个，则随机选择一个），将刚刚取出的糖果放入这堆糖果中。

现在，小 W 想请你帮他计算， K 次操作后，最终被分到糖果最多的小朋友与最少的小朋友，他们得到的糖果数之差（即操作后糖果最多的糖果堆中的糖果数与糖果最少的糖果堆中的糖果数之差）。

输入输出格式

输入格式

从文件 `candies.in` 中读入数据。

输入包含多组测试数据。

第一行包含一个正整数 T ，表示该测试点的测试数据组数。

对于每组测试数据，第一行包含两个正整数 N, K ，表示糖果的堆数和小 W 要进行的操作次数；第二行包含 N 个正整数 a_i ，表示每堆糖果初始时的糖果数。

输出格式

输出到文件 `candies.out` 中。

对于每个测试数据，输出一行包含一个正整数，表示该组测试数据的答案。

样例输入和输出

样例输入 1

```
2
4 100
1 1 10 10
4 3
2 2 2 2
```

样例输出 1

```
1
0
```

数据范围

对于 10 的数据, $N \leq 500, K \leq 40000, a_i \leq 500$;

对于 30 的数据, $N \leq 1000, K \leq 1.5 \times 10^5, a_i \leq 1000$;

对于 50 的数据, $N \leq 10000, K \leq 2 \times 10^7, a_i \leq 10^5$;

对于 100 的数据, $1 \leq N \leq 10^5, 1 \leq K \leq 1 \times 10^{15}, 1 \leq a_i \leq 10^9$ 。

B. Game(game)

题目描述

Alice 和 Bob 在玩一个游戏，Alice 和 Bob 各自控制着自己的队伍 A 和 B，假设军队中的人分布在一个一维数轴上。现在 Alice 和 Bob 每回合都可以做出一个行进指令：

以 Alice 为例，她可以让 A 队伍中的每个人都向前或向后移动 x 的距离，同理 Bob 也可以操作 B 队伍，进行一次行进。

现在告诉你战场的初始情况，即两个队伍中的人在数轴上的分布（但是我们并不知道每个人属于哪个阵营），以及经过一个回合后队伍中人的位置(只有一个回合)，你要求出 Alice 和 Bob 所做的行进指令是什么。

输入输出格式

输入格式

从文件 `game.in` 中输入数据。

第一行一个数 n ，代表两个队伍中的人的总和。

第二行 n 个整数，第 i 个整数代表第 i 个人的初始位置 x_i 。

第三行 n 个整数，第 i 个整数代表一个回合后各个人的位置 y_i 。

x_i 和 y_i 不一定对应同一个人的位置。

数据保证不会有两个人处于同一个位置。

输出格式

输出到文件 `game.out` 中。

第一行一个数 n ，代表两个队伍中的人的总和。

第二行 n 个整数，第 i 个整数代表第 i 个人的初始位置 x_i 。

第三行 n 个整数，第 i 个整数代表一个回合后第 i 个人的位置 y_i 。

数据保证不会有两个人处于同一个位置。

样例输入和输出

样例输入 1

```
5
1 2 3 4 5
70 71 0 1 72
```

样例输出 1

```
67 -1
```

Alice 控制初始的 3 4 5 三个位置，操作后为 70 71 72。

Bob 控制初始的 1 2 两个位置，操作后为 0 1。

输出两个距离代表一个可能的行进指令。数据保证至少存在一个解。

数据范围

对于 30% 的数据， $n \leq 100$ 。

对于 70% 的数据， $n \leq 1000$ 。

对于 100% 的数据， $n \leq 70000$, $0 \leq x_i, y_i \leq 100000$ 。

C. Tree(tree)

题目描述

给出一棵有 N 个点的树，每个点的编号为 $1 \sim N$ ，其中 1 号点为这棵树的根。每条边有一个权值，初始时第 i 条边的权值为 w_i 。两个点之间的距离为以这两个点为端点的简单路径上的所有边的权值和。

现在对这棵树有 M 次操作，每次操作的格式如下：

- `1 i k`: 将第 i 条边的权值修改为 k ;
- `2 x y`: 询问以 x 为根的子树中一点 u 与以 y 为根的子树中一点 v 间的最大距离;
- `3 x a b`: 询问以 x 为根的子树中一点 u 与以 a, b 为端点的简单路径上一点 v 间的最大距离;

输入输出格式

输入格式

从文件 `tree.in` 中读入数据。

第 1 行包含两个正整数 N, M , 分别表示树的结点数和操作次数;

第 2 行至第 N 行, 第 $i + 1$ 行包含三个整数 u_i, v_i, w_i , 表示第 i 条边连接节点 u_i 与节点 v_i , 初始时权值为 w_i ;

第 $N + 1$ 行至第 $N + M$ 行, 每行表示一次操作, 格式见题目描述。

输出格式

输出到文件 `tree.out` 中。

对于每次询问操作, 输出一个整数表示答案。

样例输入和输出

样例输入 1

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

样例输出 1

```
7
7
7
8
```

数据范围

测试点编号	$N, M \leq$	$w_i \leq$	$opt =$	特殊性质
1	500	1000	2, 3	
2	500	1000	1, 2, 3	
3	2000	1000	1, 2	
4	2000	1000	1, 3	C
5	2000	10^9	2, 3	A
6	2000	10^9	1, 2, 3	
7	5×10^4	1000	2	B
8	5×10^4	1000	3	A
9	5×10^4	10^9	1, 2, 3	A
10	5×10^4	10^9	1, 2, 3	B, C
11	5×10^4	10^9	1, 2, 3	
12	5×10^4	10^9	1, 2, 3	
13	10^5	10^9	1, 2	B
14	10^5	10^9	1, 3	C
15	10^5	10^9	2, 3	
16	10^5	10^9	1, 2, 3	A
17	10^5	10^9	1, 2, 3	
18	10^5	10^9	1, 2, 3	
19	10^5	10^9	1, 2, 3	
20	10^5	10^9	1, 2, 3	

对于所有测试点, $1 \leq N, M \leq 100000, 1 \leq w_i, k \leq 10^9$ 。

说明

opt 一栏表示该测试点会出现的询问种类, 与输入中给出的每次询问的第一个数字相对应。

特殊性质一栏中, A 表示该测试点给出的树是一条链; B 表示对于所有 2 操作, 都有 $x = y$; C 表示对于所有 3 操作, 都有 $a = b$ 。