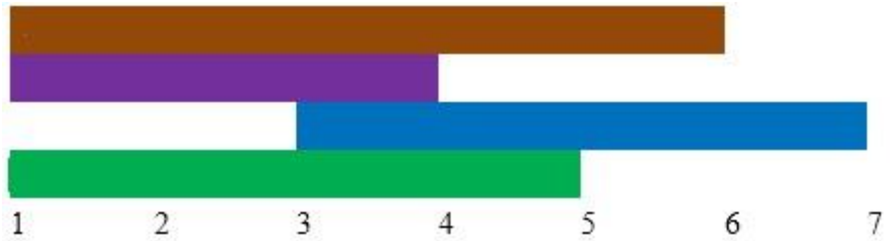
Problem 1. Wooden stack is falling down! (FINALLY)

(Time Limit: 3 seconds)

Problem Description

Linya likes playing wooden sticks whose sizes are 1cm X 1cm X u cm where u is an integer. Linya first puts one stick on the ground along a straight line, and then puts another stick on the top of the sticks along the straight line, and so on. In the progress of putting sticks on, Linya shores the stack by hand if the stack tends to fall down. After Linya puts all the sticks on the stack, Linya lets go of the stack, and wonders whether the stack falls down! Please write a program to judge it.



There are m sticks with length $a_1, a_2, a_3, \ldots, a_m$ where a_k are integers. Linya puts these sticks on the position of $x_1, x_2, x_3, \ldots, x_m$ (cm) where x_k are integers. For example, $m = 4, (a_1, a_2, a_3, a_4) = (4,4,3,5), (x_1, x_2, x_3, x_4) = (1,3,1,1)$. When we put 1^{st} stick on the position 1, and 2^{nd} stick on position 3, the center of gravity of 2^{nd} stick is supported by the edge of 1^{st} stick, and thus the stack is stable. When we put 3^{rd} stick on, it tends to fall down. But because Linya shores the stack by hand, and puts the 4^{th} stick on, the stack is stable finally.

Input Format

There is an integer n on the first line indicating the number of test cases with $n \le 20$. Each test case is written in one line as $m\ a_1\ a_2\ a_3 \ldots a_m\ x_1\ x_2\ x_3 \ldots x_m$ with $0 < m < 51, 0 < a_k < 51, 0 < x_k < 51$ for each k.

Output Format

For each test case, please output "DOWN", if the stack tends to fall down and output "STABLE" otherwise.

Example

Sample Input:	Sample Output:
2 4 4 4 3 5 1 3 1 1 3 4 3 3 2 1 3	STABLE DOWN

Problem 2. Hiring Workers

(Time Limit: 5 seconds)

Problem Description

Leo would like to hire m workers to finish n tasks. The tasks are numbered from 1 to n, and the difficulty of task i is d_i. There are m workers numbered from 1 to m, and the skill level of worker j is w_j. Worker j can handle all tasks of difficulty at most w_j, but worker j cannot handle any task of difficulty more than w_j. Leo must pay w_j^2 for hiring worker j, and each worker can do at most one task. Please write a program to compute the least amount of money to hire workers who can finish all tasks.

Technical Specification

- The number of test cases: $T \leq 30$
- $0 < n \leq m \leq 100000$
- $0 < d_i \leq 100000$ for $0 < i \leq n$
- $0 < w_j \leq 100000$ for $0 < j \leq m$

Input Format

The first line of the input contains an integer T indicating the number of test cases. Each test case has three lines. The first line contains two integers n and m separated by a blank. The number of tasks is n, and the number of workers is m. The second line consists of n integers $d_1, ..., d_n$. The difficulty of task i is d_i. The third line consists of m integers $w_1, \ldots, w_m$. The skill level of worker j is w_j.

Output Format

For each test case, output the least amount of money to hire workers who can finish the tasks. If it is impossible to finish all tasks, then output -1.

Example

Sample Input:	**Sample Output:**
2 5 6 3 1 2 4 5	55 -1

1 1 2 3 4 5 5 6 2 2 3 4 5 1 1 4 5 6 6	

Problem 3. Bisectors

(Time Limit: 3 seconds)

Problem Description

Let $T = (V, E)$ be an undirected tree. For every pair of vertices (x, y), $x, y \in V$, let $P(x, y)$ denote the unique path from x to y. An edge e is a bisector of a path H if the middle point of H is contained in e. Note that if the middle point of a path H is an internal point of an edge e, then e is the unique bisector of H. In case the middle point of H is a vertex v, then the two edges on H adjacent to v are both bisectors of H. Consider the example in Figure 1. Since the length of $P(4, 7)$ is 7, the middle point of $P(4, 7)$ is the internal point on edge (6, 9) that is at a distance 0.5 from 6 and thus edge (6, 9) is the unique bisector of $P(4, 7)$. In this example, it is easy to see that $P(5, 8)$ has two bisectors, which are edges (5, 2) and (2, 6).

For each edge e on tree T, define the *bisector count* of e, denoted by $C(e)$, as the number of vertex pairs (x, y), $x, y \in V$, $x \neq y$, such that e is a bisector of $P(x, y)$. For example, in Figure 1, since edge $e = (7, 9)$ is a bisector of $P(1, 7)$, $P(6, 7)$, $P(3, 7)$, $P(7, 8)$, $P(7, 9)$, $P(2, 7)$ we have $C(e) = 6$.

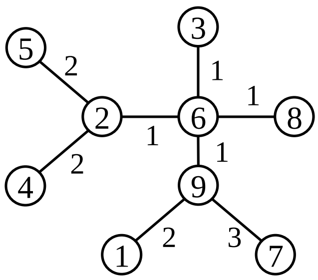


Figure 1.

Let $n = |V|$ and let $C_1, C_2, \ldots, C_{n-1}$ be the bisector counts of the edges of T. If there are k numbers, here the median is uniquely defined to be the $\lfloor (k + 1)/2 \rfloor$-th smallest number. For example, the median of (10, 5, 10, 6, 7) is 7, and the median of (2, 10, 9, 5) is 5. In this problem, you are asked to find the median and the maximum of $C_1, C_2, \ldots, C_{n-1}$. For example, if the bisector counts are 10, 5, 10, 6, 7, the output is 7 and 10.

Technical Specification

- The number of vertices n is a positive integer between 2 and 3000.
- The length of each edge is a positive integer between 1 and 1000.

Input Format

The first line is an integer t, $1 \le t \le 10$, indicating the number of test cases. Each test case starts with one line containing a number n, $2 \le n \le 3000$, indicating the number of vertices in the tree T. Then, $n - 1$ lines follow, each of which contains three integers x, y, l, $1 \le x$, $y \le n$, $1 \le l \le 1000$, indicating there is an edge of length l connecting vertices x and y.

Output Format

For each test case, output one line containing two numbers c_1 and c_2, where c_1 and c_2, respectively, are the median and maximum of the $n - 1$ bisector counts.

Example

Sample Input:	**Sample Output:**
2 4 1 2 2 1 3 2 1 4 3 9 5 2 2 4 2 2 2 6 1 6 3 1 6 8 1 6 9 1 9 1 2 9 7 3	2 3 6 12

Problem 4. Filling Letters

(Time Limit: 2 seconds)

Problem Description

Suppose there is a rooted tree T and each leaf node of T is associated with an English letter. You need to assign an English letter for every internal node. The cost of your assignment is defined as the total number of tree edges whose two ends are assigned with different letters. In Figure 1 for example, if we assign A to v_1, A to v_2 and B to v_3, the cost would be 2, which comes from edges (v_1, v_3) and (v_3, v_8). Your job is to find an optimal assignment that can reach the minimum cost.

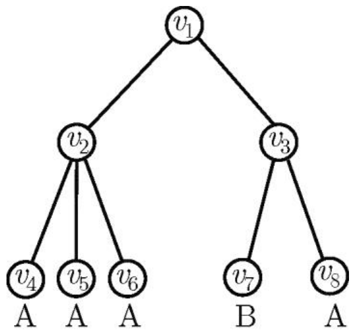


Figure 1.

Technical Specification

- There are at most 10 test cases.
- The tree nodes are labelled by numbers 1 to n where $2 \le n \le 10000$. Node 1 is always the root. A leaf node is a node that has no child, and its label may be any number between 2 to n.
- The leaf nodes are assigned with only capital English letters.

Input Format

The first line of the input contains an integer indicating the number of test cases. Each test case starts with one line containing a number n, where $2 \le n \le 10000$, indicating the number of nodes in the tree. Then n-1 lines follow, and each contains two integers i and j where there is an edge between nodes i and j. The nth line is a string of capital English letters that gives you the assignment of letters for the leaf nodes in ascending order by their labels, respectively.

Output Format

For each test case, please output the minimum cost of the specified tree.

Example

Sample Input:	**Sample Output:**
2 3 1 2 1 3 AB 3 1 2 2 3 G	1 0

Problem 5. Sharing a Donut

(Time Limit: 2 seconds)

Problem Description

Jack is one of Joy's good friends. They would like to share a circular donut. The donut can be divided into n parts. These parts are numbered from 1 to n in clockwise order, and the deliciousness of part i is d_i. They plan to cut the donut into two pieces such that the difference of total deliciousness of two pieces is minimized. Could you help them to compute?

Technical Specification

- The number of test cases: $T \leq 10$
- $1 < n \leq 1000000$
- $0 < d_i \leq 100$ for $0 < i \leq n$

Input Format

The first line of the input contains an integer T indicating the number of test cases. Each test case has two lines. The first line contains an integer n indicating the number of parts. The second line consists of n integers $d_1, \ldots, d_n$. The deliciousness of part i is d_i.

Output Format

For each test case, output the total deliciousness of two pieces, t_1 and t_2, such that their difference is minimized. You should choose $t_1 \leq t_2$ and separate them by a blank.

Example

Sample Input:	**Sample Output:**
2 5 3 1 2 4 5 5 1 1 5 1 1	6 9 2 7