K. N. Toosi University of Technology

Algorithm Design Course
Instructor : Dr. Shafiei
Written By TA : Sepehr Shahsavar

Ali and The Haghs

Jun , 2021

Overview

Ali often uses his smartphone. He has already downloaded n haghs on it. hagh with number i takes up a_i units of memory.

Ali wants to free at least m units of memory (by removing some haghs).

Of course, some haghs are more important to Ali than others. He came up with the following scoring system — he assigned an integer b_i to each hagh:

- $b_i = 1$ — regular hagh.
- $b_i = 2$ — important hagh.

According to this rating system, his phone has $b_1 + b_2 + \ldots + b_n$ convenience points.

Ali believes that if he removes haghs with numbers $i_1, i_2, \ldots, i_k$, then he will free units of memory and lose $b_{i_1} + b_{i_2} + \ldots + b_{i_k}$ convenience points.

For example, if n=5, m=7, a=[5,3,2,1,4] , b=[2,1,1,2,1], then Ali can remove the following hagh sets (not all options are listed below):

- haghs with numbers 1,4 and 5. In this case, it will free $a_1 + a_4 + a_5 = 10$ units of memory and lose $b_1 + b_4 + b_5 = 5$ convenience points;
- haghs with numbers 1 and 3. In this case, it will free $a_1 + a_3 = 7$units of memory and lose $b_1 + b_3 = 3$convenience points.
- haghs with numbers 2 and 5. In this case, it will free $a_2 + a_5 = 7$ memory units and lose $b_2 + b_5 = 2$ convenience points.

Help Ali , choose a set of haghs, such that if removing them will free at least m units of memory and lose the minimum number of convenience points, or indicate that such a set does not exist.

2

Input

The first line contains one integer t ($1 \le t \le 10^4$) — the number of test cases. Then t test cases follow.

The first line of each test case contains two integers n and m ($1 \le n \le 2 * 10^5$, $1 \le m \le 10^9$) — the number of haghs on Ali's phone and the number of memory units to be freed.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 2$) — the number of memory units used by haghs.

The third line of each test case contains n integers $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le 2$) — the convenience points of each hagh.

It is guaranteed that the sum of n over all test cases does not exceed 2 · 105.

Output

For each test case, output on a separate line:

- `-1`, if there is no set of haghs, removing which will free at least m units of memory;
- the minimum number of convenience points that Ali will lose if such a set exists.

Example

Input :

5

5 7

5 3 2 1 4

2 1 1 2 1

1 3

2

1

3

5 10

2 3 2 3 2

1 2 1 2 1

4 10

5 1 3 4

1 2 1 2

4 5

3 2 1 2

2 1 2 1

Output :

2

-1

6

4

3

Note

In the first test case, it is optimal to remove haghs with numbers 2 and 5, freeing 7 units of memory. $b_2 + b_5 = 2$

In the second test case, by removing the only hagh, Ali will be able to clear only 2 of the memory units out of the 3 needed.

In the third test case, it is optimal to remove haghs with numbers 3 and4, freeing 10 units of memory. $b_1 + b_2 + b_3 + b_4 = 6$.

In the fourth test case, it is optimal to remove haghs with numbers 1, 3 and 4, freeing 12 units of memory. $b_1 + b_2 + b_4 = 4$

In the fifth test case, it is optimal to remove haghs with numbers 1 and 2, freeing 5 units of memory. $b_1 + b_2 = 3$.

Goals :

- Design a Dynamic Programming Algorithm that can help Ali remove the haghs .

YOUR APPROACH WILL NOT BE ACCEPTED IF IT'S NOT DYNAMIC PROGRAMMING !