Rainbow Bracket Sequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

A bracket sequence is a string containing only characters `(` and `)`. A regular bracket sequence is a bracket sequence that can be transformed into a correct arithmetic expression by inserting characters `1` and `+` between the original characters of the sequence. For example, bracket sequences `()()` and `(())` are regular, and `)(`, `(`, and `)` are not.

There are m different colors, and you are given $\ell_1, \ldots, \ell_m$. Consider bracket sequences of length $2n$, the colors of each position are $c_1, \ldots, c_{2n}$. A regular bracket sequence is *colorful* if

- For all $i = 1, \ldots, m$, denote t_i to be the number of positions of color i with left bracket `(` on it, then $t_i \geq \ell_i$ holds.

Given values of positions $v_1, \ldots, v_{2n}$, the *value* of a regular colorful bracket sequence is the sum of values of the positions with left bracket.

You need to find the maximum value among all regular colorful bracket sequences. If there is no such sequence, output `-1` instead.

Input

Each test contains multiple test cases. The first line contains the number of test cases $t (1 \leq t \leq 100)$. The descriptions of the test cases follow.

Each test case contains four lines. The first line contains two integers $n, m (1 \leq n \leq 100, 1 \leq m \leq n)$, indicating the half of the length of the bracket sequence and the number of colors.

The second line contains m integers $\ell_1, \ldots, \ell_m (0 \leq \ell_i \leq n)$, indicating the limit of each color. The third line contains $2n$ integers $c_1, \ldots, c_{2n} (1 \leq c_i \leq m)$, indicating the color of each position. And the fourth line contains $2n$ integers $v_1, \ldots, v_{2n} (1 \leq v_i \leq 10^9)$, indicating the value of each position.

It is guaranteed that the sum of n over all test cases does not exceed 500.

Output

For each test case, print one line with one integer indicating the answer to the question or `-1` if there is no possible sequence.

Example

standard input	standard output
2 3 2 1 2 1 2 2 2 1 2 3 1 4 2 2 1 3 2 2 2 1 2 2 2 1 2 3 1 4 2 2 1	9 -1

Note

In the first example, sequence `()(())` gets the maximum 9. And in the second example, there is no possible sequence since there are only 3 left brackets.