B: Little Engine that Could

In the classic story of *The Little Engine that Could* the engine finally succeeded in pulling the train over a mountain. You will be presented with multiple cases in which the little engine attempts to pull a train of cars over simulated mountains, for which you will determine if it succeeds or fails. The engine has a given amount of traction on the track. If the pull of the cars back down the mountain exceeds this traction, the engine fails. By rearranging the cars, it might be possible to get a few lighter cars over the top to help counterweight heavier cars that have not yet reached the slope. You will not have to determine if rearranging can help, but keep it in mind as you attempt to solve the problem.

Measurements will be given in meters for lengths of cars, which will all be the same for a given train. The slope will be given as a ratio of rise to horizontal distance. Slopes between pairs of designated points are assumed to be constant. Traction and weights will be given in metric tons. Recall from your physics that the backward force of a rail car on a slope is simply the weight times the slope.

Input

There may be multiple cases. The number of cases is indicated as an integer on the first line of input. Each case is presented on three lines. The first of these lines contains the number of cars including the engine, the length of each car, the traction (positive value) of the engine, and the number of slope segments (at least 1) representing the mountain. The second line contains a sequence of positive numbers representing the weights of the cars, ordered from front to back, starting with the engine. This includes the weight of the engine inself. The third line contains pairs of numbers representing the slope of a segment and the length of that segment. Slopes may be negative (sloping downward). Lengths will be given a (integral) multiples of car lengths. (This ensures that all cars go over the points where the slope may change at the same time.) You may assume the slope is 0 for portions of the track before encountering the mountain and after leaving the mountain and there is plenty of track.

Output

For each case, display the case number and the answer as either *yes* or *no*.

Sample Input

```
2
1 50 100 2
90
0.5 2   0.4 3
3 20 49 3
40 40 20
0.4 3   0.5 3   -0.3 2
```

Sample Output

```
Case 1: yes
Case 2: no
```