

Frequent Number in Time Window

We prefer a solution in Java, but Python or C++ would be acceptable. You have two hours to complete this challenge. Please e-mail us with any questions. Good luck!

You are given a sequence $L = \langle (a_0, t_0), (a_1, t_1), (a_2, t_2), \dots, (a_{n-1}, t_{n-1}) \rangle$ of pairs of integers with $t_0 < t_1 < t_2 < \dots < t_{n-1}$, and positive integers W and B . Given an integer t , define S_t to be the *multiset* (i.e., with multiplicities) of integers a_i with $t \leq t_i < t + W$.

Write a method that takes L, W, B and outputs a Boolean which is true if and only if there is an integer t such that some element in S_t occurs at least B times.

Example 1: In $L = \langle (2, -2), (-1000, 1), (7, 3), (7, 4), (4, 6), (7, 7), (9, 8), (1, 9), (7, 500000000) \rangle$ with $W = 5$ and $B = 3$, the answer is "true" because for $t = 3$, $S_t = S_3 = \{7, 7, 4, 7\}$ (remember that S_t is a *multiset*), which contains $B = 3$ copies of 7.

Example 2: In the same input but with $B = 4$, the answer is "false" because there is no window of size $W = 5$ which contains $B = 4$ copies of any integer. For example, while 7 does appear four times overall, there is no window of size $W = 5$ with $B = 4$ or more 7's. Specifically, $S_3 = \{7, 7, 4, 7\}$ (which contains no integer of multiplicity at least 4); $S_2 = \{7, 7, 4\}$, which likewise contains no integer of multiplicity at least 4; $S_1 = \{-1000, 7, 7\}$ (likewise), and each of the other infinitely many S_t 's contains no integer of multiplicity 4 or more.

You may assume that W and B appear first, on one row, followed by the n pairs $a_i t_i$ appearing on separate rows. Hence the input file for Example 1 would be:

```
5 3
2 -2
-1000 1
7 3
7 4
4 6
7 7
9 8
1 9
7 500000000
```

Your code will be judged on correctness, efficiency, and style. Make sure you include comments so that we can understand what you've done.