



Module: Streaming Algorithms (Week 5 out of 5)
Course: [Advanced Algorithms and Complexity](#) (Course 5 out of 6)
Specialization: [Data Structures and Algorithms](#)

Programming Assignment 5: Streaming Algorithms

Revision: October 30, 2017

Introduction

Welcome to the final (but optional) programming assignment of the [Advanced Algorithms and Complexity class](#)! In this assignment, you will practice implementing CountSketch algorithm covered in the lectures.

1 Problem: Nice vs Naughty



It's Christmas time, and Santa Claus is preparing presents for children again this year. This year he has a new strategy for allocating presents, and what each child gets depends on whether they were *nice* or *naughty*. Santa has two lists: list A contains the number of times each child did something good throughout the year, and List B contains the number of times each child did something bad. Santa has fixed a threshold t for this year and will consider a child nice if the number of good things they did is at least t larger than the number of naughty things they did during the year. Sounds simple enough, but Santa wants to make sure that his helper elves can quickly tell him whether a given child was nice or naughty, so he posed a challenge to them to test their skills.

Santa will read the two lists to the elves first, and only then he will give them the names (ids) of the children he is interested in and ask whether these children were nice or naughty. This makes sense, as the lists are long, so he only wants to read them to the elves once, and get answers quickly later (Santa is a bit lazy this year). If the elves could memorize the lists the challenge would be easy, but they have short memory. They tried to convince Santa to give them the child ids that he is interested first and then the lists (this would of course make their task easy – they could ignore all but the children he is interested in when he reads the lists), but he refused. They need your help!

List A is a sequence of integer tuples $(k_i, g_{k_i}), i = 1, \dots, N$, where for each child $k_i, i = 1, \dots, N$ the number of good things done by child with id k_i throughout the year equals g_i . Similarly, list B is a sequence of integer tuples $(k_i, b_i), i = 1, \dots, N$, where for each child $i = 1, \dots, N$ the number of bad things done by child with id k_i throughout the year equals b_i . Santa first tells the elves the number of children N , and the positive integer threshold $1 \leq t \leq 10^6$ that he is interested in. Santa first reads list A to the elves once, then list B . After that he gives the elves a small set of child ids k_1, \dots, k_r , and asks them which of the children $k_i, i = 1, \dots, r$ were nice and which were naughty, i.e. for each i whether $g_{k_i} - b_{k_i} \geq t$ or $g_{k_i} - b_{k_i} < t$.

Problem Description

Input Format. The first line contains the number of children N , the second line contains the threshold t . This is followed by list A : N lines specifying the number of good things each child did during the year. For each $i = 1, \dots, N$ the corresponding line contains the tuple (k_i, g_{k_i}) (separated by a space). The next N lines specify list B : the number of bad things each child did during the year. For each $i = 1, \dots, N$ the corresponding line contains the tuple (k_i, b_{k_i}) . The next line contains the number of queries q . The last line contains q children ids.

Constraints. All the number are integers. $1 \leq N \leq 5 \cdot 10^5$; $0 \leq k_i \leq 2^{31} - 1$; $0 \leq b_j, g_j \leq 2^{31} - 1$, $1 \leq q \leq 10$. It is guaranteed that the lists A and B contain the same children ids (though not necessarily in the same order), and that each id appears exactly once in A and exactly once in B . All the query ids are present in A and B .

Output Format. For each query id, output 1 if this child is nice, and 0 if the child is naughty. Separate these q bits by spaces.

Time and Memory Limits.

language	C	C++	Java	Python	C#	Haskell	JavaScript	Ruby	Scala
time (sec)	4	4	40	240	6	8	20	20	12
memory (Mb)	10	10	300	50	10	10	10	10	10

Sample 1.

Input:

```
3
2
3 42
8 50001
11 230040
8 50000
3 40
11 230040
2
8 3
```

Output:

```
0 1
```

There are three children: with ids 3, 8, and 11. Child 8 is naughty since $50001 - 50000 < 2$, while child 3 is nice since $42 - 40 \geq 2$.

Starter Files

The starter solutions for this problem implement a naive solution that just stores everything in a map (and hence uses too much memory). Filename: `heavy_hitters`

Need Help?

Ask a question or see the questions asked by other learners at [this forum thread](#).