ACM Algorithm Practice **Spring 2015**

Week 2 Problems

April 7, 2015

Timothy Johnson

Problem 1: Photo to Remember

(Taken from http://codeforces.com/problemset/problem/522/B)

One day n friends met at a party. They hadn't seen each other for a long time, and so they decided to make a group photo together.

Simply speaking, the process of taking photos can be described as follows. On the photo, each photographed friend occupies a rectangular of pixels: the i-th of them occupies a rectangle of width w_i pixels and height h_i pixels. On the group photo everybody stands in a line, so the minimum pixel size of the photo is $W \times H$, where W is the total sum of all widths and H is the maximum height of all the photographed friends.

As is usually the case, the friends made n photos. The j-th ($1 \leq j \leq n$) photo had everybody except for the j-th friend, who was the photographer.

Print the minimum size of each photo in pixels.

Input:
The first line contains an integer n ($1 \leq n \leq 200000$) - the number of friends.

Then n lines follow. The i-th line contains a pair of integers w_i, h_i ($1 \leq w_i \leq 10, 1 \leq h_i \leq 1000$), which are the width and height of the rectangle containing the i-th friend in pixels.

Output:
Print n space-separated numbers $b_1, b_2, \ldots, b_n$, where b_i is the total number of pixels on the smallest photo containing every friend except for the i-th one.

Examples:

- **Input:**
  3
  1 10
  5 5
  10 1

  Output:
  75 110 60

- **Input:**
  3
  2 1
  1 2
  2 1

  Output:
  6 4 6

Problem 2: Number of Ways

(Taken from http://codeforces.com/problemset/problem/466/C)

You have an array $a[1], a[2], \ldots, a[n]$ consisting of n integers. Count the number of ways to split all the elements of the array into three contiguous parts so that the sum of the elements in each part is the same.

More formally, you need to find the number of pairs of indices $i, j (2 \le i \le j \le n-1)$ such that $\sum_{k=1}^{i-1} a_k = \sum_{k=i}^{j} a_k = \sum_{k=j}^{n} a_k$.

Input: The first line contains integer $n (1 \le n \le 5 * 10^5)$, showing how many numbers are in the array. The second line contains n integers $a[1], a[2], \ldots, a[n] (|a[i]| \le 10^9)$, the elements of array a.

Output: Print a single integer, the number of ways to split the array into three parts with the same sum.

Examples:

- **Input:**
  5
  1 2 3 0 3

  Output:
  2

- **Input:**
  4
  0 1 -1 0

  Output:
  1

- **Input:**
  2
  4 1

  Output: 0

Problem 3: Clique Problem

(Taken from http://codeforces.com/problemset/problem/527/D)

The clique problem is one of the most well-known NP-complete problems. Under some simplification it can be formulated as follows. Consider an undirected graph G. It is required to find a subset of vertices C of the maximum size such that any two of them are connected by an edge in graph G. Sounds simple, doesn't it? Nobody yet knows an algorithm that finds a solution to this problem in polynomial time of the size of the graph. However, as with many other NP-complete problems, the clique problem is easier if you consider a specific type of a graph.

Consider n distinct points on a line. Let the i-th point have the coordinate x_i and the weight w_i. Let's form graph G such that the vertices are the points and there is an edge between vertex i and vertex j if and only if the distance between then is at least the sum of their weights: $|x_i - x_j| \geq w_i + w_j$.

Find the size of the maximum clique in such a graph.

Input: The first line contains the integer n ($1 \leq n \leq 200000$), the number of points.

Each of the next n lines contains two numbers x_i, w_i ($0 \leq x_i, w_i \leq 10^9$) - the coordinate and the weight of a point. All x_i are different.

Output: Print a single number - the number of vertices in the maximum clique of the given graph.

Example:

- **Input:**
  4
  2 3
  3 1
  6 1
  0 2

  Output:
  3