Practice S12P07: Pixels (Speed Challenge!)

http://www.comp.nus.edu.sg/~cs1010/4_misc/practice.html

Week of release: Week 12

Objective: Problem solving

(Note: This exercise is the same as S12P06 except that the test cases are larger. You will need to come up with an efficient algorithm to pass all the test cases.)

Task statement:

The most common way of representing pixels (picture elements) is to use the RGB (red, green and blue) values, which range from 0 to 255.

Given a list of pixels represented by their RGB values, we may define similarity as follows:

A list is of similarity measure ***D*** if for all pixels in this list,

$$A\times(r - r_{min}) + B\times(g - g_{min}) + C\times(b - b_{min}) \le D$$

where A, B, C and D are integer constants, r, g, and b are the RGB values for the pixel, and r_{min}, g_{min}, and b_{min} are the minimum R-, G-, and B-values among all the pixels in this list.

(Note: More than one pixel may have the same RGB value.)

Given the RGB values of n pixels, and the integer constants A, B, C, and D, we want to find the size of the largest sublist of pixels with similarity D.

Write a program **pixels.c** to solve this problem.

The first line of the input contains 5 integers: n, A, B, C, and D (where $1 \le n \le 20$, $1 \le A, B, C \le 10000$, and $1 \le D \le 10000000$).

The following n lines each contains three integers r, g and b (where $0 \le r, g, b \le 255$) representing the RGB values of a pixel.

Your program should output the size of the largest sublist of similarity measure D.

Example

For example, given a list of 5 pixels:

(3,7,5), (4,8,6), (5,9,7), (4,9,7), (8,9,6),

and given also that $A = B = C = 1$, and $D = 5$,

if we pick a sublist that contains the first, second and fourth pixels {(3,7,5), (4,8,6), (4,9,7)}, then $r_{min} = 3$, $g_{min} = 7$, and $b_{min} = 5$.

For the first pixel (3,7,5), the measure is 1×(3-3) + 1×(7-7) + 1×(5-5) = 0. For the second pixel (4,8,6), the measure is 1×(4-3) + 1×(8-7) + 1×(6-5) = 3. For the fourth pixel (4,9,7), the measure is 1×(4-3) + 1×(9-7) + 1×(7-5) = 5.

Since all the measures are less than or equal to *D* (5), this sublist of 3 pixels satisfies the similarity requirement. However, this is not the largest sublist possible. if we pick a sublist that contains the first, second and fourth pixels {(3,7,5), (4,8,6), (4,9,7)}, then r_{min} = 3, g_{min} = 7, and b_{min} = 5.

The largest sublist that satisfies the similarity requirement consists of the second, third, fourth and fifth pixels. (Verify this yourself.)

Since the size of this sublist is 4, and we cannot find a larger sublist that satisfies the similarity requirement, the answer is 4.

Sample run:

```
5 1 1 1 5
3 7 5
4 8 6
5 9 7
4 9 7
8 9 6
4
```