Programming Assignment 2: Foreman

Darin Goldstein

1 Deadline

Friday, October 8th at 5 PM

2 Introduction

A foreman wants to put together a team to build a monument. The head of HR has split the available workforce into two main categories: the A's and the B's. If a worker is in category A, then he is able to complete $a > 0$ units of work per hour; similarly, a worker in category B can complete $b > 0$ units of work per hour.

The head of HR is both super-competent but extremely forgetful: (super-competent) HR has put together a list of potential teams to complete the project and the foreman is able to choose any team he wants. A team consists of two nonnegative integers (x_a, x_b) where x_a is the number of category A people on the team and x_b is the number of category B people on the team. (extremely forgetful) Even though HR has put forward this list of potential teams, HR has not informed the foreman of the actual values of a and b. The foreman has no knowledge of which of the two is even larger than the other. He only knows that because HR is not a total idiot, $a \neq b$. You may assume that HR is on vacation somewhere and is unreachable for this information.

Given this list of potential teams, the foreman calls you in for a simple assignment. Though there are a potentially infinite number of possibilities for the values of a and b, there are only a finite number of ways to arrange in increasing order the total amount of work that the team can accomplish. For example, if the list looks like $(1, 1), (2, 1), (1, 2)$, then there are only two possible sorted lists: $(1, 1), (2, 1), (1, 2)$ or $(1, 1), (1, 2), (2, 1)$. The foreman wants you to calculate all of the valid possibilities for sorted lists.

3 Your code

You will write a class StudentSolver that determines all possible sorted lists, given the list of possible worker combinations.

If you are writing the file in Java: StudentSolver.java should have a function with the header `public static ArrayList<ArrayList<Pair<Integer,Integer>>> solve(ArrayList<Pair<Integer,Integer>> list)`

If you are writing the file in Python: studentsolver.py should have a function with the header `def solve(problem)`

If you are writing the file in C++: StudentSolver.h should have a line with the header `static std::vector<std::vector<std::pair<int, int>>> solve(const std::vector<std::pair<int, int>>& list);`

4 Example

Consider the following example: (1,2), (2,1), (2,4), (4,2).

There are only two valid sorted orders: (1, 2), (2, 1), (2, 4), (4, 2) and (2, 1), (1, 2), (4, 2), (2, 4).

Notice that the ordering (1, 2), (2, 1), (4, 2), (2, 4) is invalid because if $(1,2) = a + 2b$ is strictly less than $(2,1) = 2a + b$, then $(2,4) = 2(a + 2b)$, which is exactly twice as heavy as (1, 2), must be strictly less heavy than $(4,2) = 2(2a + b)$, which is exactly twice as heavy as (2, 1).

In other words, with the ordering of $(1,2),(2,1),(4,2),(2,4)$ we see that $(1,2) = a+2b$ is strictly less than $(2,1) = 2a+b$. We then have $(4,2) = 4a+2b = 2(2a+b) = 2*(2,1)$ is strictly less than $(2,4) = 2a+4b = 2(a+2b) = 2*(1,2)$. So if $(1,2)$ is strictly less than $(2,1)$, how can $2*(2,1)$ be strictly less than $2*(1,2)$? This would create a contradiction and yield an invalid ordering.