

Individual Project: Delivering Bundles

William Shoaff

CSE 5211 Analysis of Algorithms

Spring 2019 (December 4, 2018) Due: Tuesday, February 22, 2019

The Task

Feeling greedy and mischievous Pablo Escobar decided to start running drugs in Brevard county. He enlisted Larrico Smith to help him stock pile bundles of marijuana at docks along the Indian River Lagoon.

There are N ($1 \leq N \leq 1,000,000$) docks conveniently numbered $1 \dots N$, initially all of them have no bundles. Escobar then gives Larrico a sequence of K instructions ($1 \leq K \leq 25,000$), each of the form “A B”, meaning that Larrico should add one new marijuana bundle to each dock in the range A to B (inclusive). For example, if Larrico is told “10 13”, then he should add a bundle of marijuana to each of the docks 10, 11, 12, and 13.

After Larrico finishes, Escobar would like to know how evenly distributed the bundles are. Please help Larrico determine the answer to Escobar’s question.

Sample Input

On the first line are two, space-separated integers, N and K . Each of the next K lines contain one of Escobar’s instructions in the form of two, space-separated integers A B where ($1 \leq A \leq B \leq N$). For example,

7	5
1	4
5	5
2	4
4	6
3	5

In this sample input there are $N = 7$ docks, and Escobar issued $K = 5$ instructions:

The first instruction is to add a bundle of marijuana to docks 1 through 5, The second is to add a bundle of marijuana to dock 5, and so on.

Sample Output

The output is one number, the difference between the largest number of bundles and the smallest number of bundles.

In the sample input, After Larrico is finished, the docks have 1, 3, 3, 4, 3, 3, bundles. In the sorted order: 1, 3, 3, 3, 3, 4 we can see the difference between the dock with the most and the dock with the least is $4 - 1 = 3$ (three) bundles.

The program

Write a **HASKELL** program that computes the difference in bundles between the dock with the most and the dock with the least bundles.

Data Structures

Storing the data in lists and updating (mutating) the lists as Escobar gives instructions is a natural solution strategy, but it may not be efficient.

Write and profile a Haskell list-based program that computes the number.

Direct access arrays afford a more efficient solution strategy. The Haskell module `Data.Array.Iarray` provides an interface to immutable (once created cannot be changed), strict (Need a cogent explanation here) The `Ix` class that supports array indexing is imported along with `Data.Array.Iarray`.

Write and profile a Haskell array-based program that computes the number.

The class `Data.Array.MArray` supports an interface to mutable arrays.

Write and profile a Haskell mutable array-based program that computes the number.

Testing and profiling your program

Test Data

You will need to test your programs. Mimic the Sample Data to create a few files containing short lists of bundle delivery instructions where number can be hand-calculated and use these to boost your confidence in the correctness of your code.

You will want to stress your code to observe how it behaves asymptotically. Recall there are 1 million docks and Escobar can issue up to 25,000 instructions "A B" indicating where to drop bundles.

Write a program, in any language, that will create a file of random instructions. You will want the number of instructions to grow in some pattern.

Profiling

Consult [The Glasgow Haskell Compiler User's Guide on Profiling](#) to learn how to compile your programs so that profiling reports are generated when your code executes.

Using the profile data, write a summary report on the performance for the different solution implementations.

Turning in your work

By the due date, turn in your source programs, and report on your execution profiles.

It is important that your programs be reasonably commented. I recommend [Knuth's Literate Programming](#) and tools that support it.

At the beginning of your source file include the following header information in comments:

```
# Author: name, e-mail address
# Course: CSE 5211, Spring 2019
# Project: Marijuana Bundles
```

Use the [the submission server](#) to turn in the **HASKELL** source code for your programs, a description of your tests, and your report comparing the performance of the different solutions.

You will need a control code from the [submit server](#) in order to submit. After registering the submit server will e-mail the control code to you. The file name to submit must be `bundles.hs` and the project id is `bundles`.