
CpSc 2120: Algorithms and Data Structures

Instructor: Dr. Brian Dean

Webpage: <http://www.cs.clemson.edu/~bcdean/>

Handout 9: Homework Assignment #2

Fall 2021

MWF 9:05-9:55

Powers 208

1 Three Algorithmic Challenges

This assignment will give you practice using combinations of the algorithmic tools we have learned so far, particularly binary search trees and sorting. The assignment consists of three separate parts. For each, you'll need to submit a program running in $O(n \log n)$ time or faster for full credit.

2 Problem 1: Finding Good Football Teams

Input to this problem is given by n lines on standard input, each describing a football team. Each line contains three space-separated items: the name of the team (a string using lowercase characters only), the team's offensive strength, and the team's defensive strength. Offensive and defensive strengths are nonnegative integers each at most 1 billion. For example:

```
clemson 1 10
ncstate 5 11
georgia 6 7
```

Team A is *in trouble* if there exists some other team B for which B has strictly larger offensive *and* defensive strength than A . In the example above, “clemson” is in trouble because “ncstate” surpasses it both in offense and defense. Please print out the names of all the teams that are *not* in trouble, in sorted order. In the example above, that would be “ncstate” and “georgia”.

3 Problem 2: Breaking Prof. Dean's Quicksort

In the directory

`/group/course/cpsc212/f21/hw02/`

you will find a program `quicksort.cpp` written by Prof. Dean, which implements the quicksort algorithm. It accepts input from standard input in the following format: an integer n , followed by n non-negative integers. The maximum value of n is 100,000.

Please submit a program that reads n from standard input and prints to standard output an input of size n that when fed to Prof. Dean's code makes it run in $\Omega(n^2)$ time (this should translate to several seconds of running time on the lab machines for $n = 100,000$). Note that your code should still generate this bad input in only $O(n \log n)$ time.

4 Problem 3: The 2120 Shuffle

The “2120 shuffle” is the hottest new trend in dancing! An example transcript of this dance is given to you as n lines of input on standard input. Each line contains two space-separated strings x and y (each using only lowercase letters). For example:

```
serena hayden
patrick sydney
sydney serena
hayden patrick
```

All of the individuals mentioned in this input (4 of them, in this example) start the dance lined up in sorted order. Each line then specifies the order in which individuals move to new positions in the line. A line of input “ $x y$ ” means that person x moves to the position directly after person y . So in our example, the dance proceeds as follows:

hayden patrick serena sydney	(initial ordering)
hayden serena patrick sydney	(first, serena moves to the position after hayden)
hayden serena sydney patrick	(then patrick moves to the position after sydney)
hayden serena sydney patrick	(sydney doesn’t move, being already right after serena)
serena sydney patrick hayden	(finally hayden moves to the position after patrick)

When a person actively moves from index i to index j in the line, this takes $|i - j|$ work (note that other people whose indices change on account of another person actively moving don’t spend any work). Please print out the amount of work spent by each person, in alphabetical order. In our example:

```
hayden 3
patrick 1
serena 1
sydney 0
```

5 Submission and Grading

Submit your code using `handin.cs.clemson.edu`. Zero points will be awarded for code that does not compile, so make sure your code compiles on the lab machines before submitting! If any of your programs consist of multiple files, please submit a makefile so that typing “make” will compile all three programs.

Final submissions are due by 11:59pm on the evening of Wednesday, October 13. No late submissions will be accepted.