

---

# CpSc 2120: Algorithms and Data Structures

**Instructor:** Dr. Brian Dean

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

**Handout 8:** Homework #2

Fall 2019

MW 2:30-3:45

Daniel 313

---

## 1 It's Fun to be the Bad Guy...

In this homework, you will play the role of a evil, malicious adversary whose goal in life is to make programs run slowly.

Files for this assignment are located here::

`/group/course/cpsc212/f19/hw02/`

In this directory, you will find three programs: `prog1.cpp`, `prog2.cpp`, and `prog3.cpp`. Each of these accepts input from standard input in the same format: an integer  $N$ , followed by  $N$  non-negative integers. The maximum value of  $N$  is 100,000.

Your goal is to carefully examine these three programs, and to design inputs for them that will cause them to run slowly. Moreover, you are to write three programs `bad1.cpp`, `bad2.cpp`, and `bad3.cpp`, that each respectively generate bad inputs for `prog1.cpp`, `prog2.cpp`, `prog3.cpp`. Each of your programs should take a single argument on the command line giving the input size, and it should print to standard output a bad input case of that size. You are not allowed to change `prog1.cpp`, `prog2.cpp`, or `prog3.cpp`. An example to get you started with `bad1.cpp` is provided – this doesn't generate a very difficult input for `prog1.cpp` yet though, so you'll still need to make some modifications.

## 2 Testing

To test your code, you could for example run the following:

```
g++ -o prog1 prog1.cpp
g++ -o bad1 bad1.cpp
time ./bad1 100000 > input1
time ./prog1 < input1
```

For `prog3`, you may also want to re-direct the output to a file, since there is a large amount of output, so that the time spent printing the output does not obscure the overall running time:

```
time ./prog3 < input1 > output1
```

### 3 Running Time Goals

You should make each of the sample programs `prog1.cpp`, `prog2.cpp`, and `prog3.cpp` take  $\Omega(N^2)$  time, which should translate to 3 seconds or more (possibly quite a bit more) for large input cases; the exact running time will depend on the machine you use, of course.

Your programs `bad1.cpp`, `bad2.cpp`, and `bad3.cpp` should run very quickly, in at most  $O(N \log N)$  time each, which should translate to well under one second even for the largest input cases.

### 4 Submission and Grading

Please submit your three programs `bad1.cpp`, `bad2.cpp`, and `bad3.cpp` using `handin.cs.clemson.edu`, just as with the lab assignments. Your assignment will be graded based on correctness, and also on the clarity and organization of your code. Final submissions are due by 11:59pm on the evening of Sunday, October 6. No late submissions will be accepted.