ECE 592 – Topics in Data Science

Test 2: Scientific Programming – Fall 2022

October 3, 2022

Please remember to justify your answers carefully.

Last name: _____ First name: _____

Please recall the course academic integrity policy (from the syllabus):
When working on tests, no cooperation or "collaboration" between students is allowed. While it could be tempting to text or email a friend during a test that is administered electronically, this is not allowed. You will be allowed to use your notes, books, a browser, and software such as Matlab and/or Python.[1] However, while working on the test you should not text, email, or communicate with other people (certainly not other students) in any way, unless you are consulting with the course staff. By submitting the test, you will be acknowledging that you completed the work on your own without the help of others in any capacity. Any such aid would be unauthorized and a violation of the academic integrity policy.

---

[1]You can use the browser to access Moodle, the course webpage, and look up technical topics. Similar to a normal test, you must not communicate with other people.

Question 1 (Computational complexity.)
What is the computational complexity of the following pseudo code? Express your answer as a function of N using $\Theta(\cdot)$ notation. Make sure to justify your answer.

```
a = 0 ## initialize the variable a with 0
for i = 1 to N { ## loop over i
    for j = 1 to ceiling(square_root(i)) ## loop over roughly sqrt(i) values
        a = a + my_function(i,j) ## increment by function (runs in constant time)
    } ## end loop over j
} ## end loop over i
```

Question 2 (Algorithms.)

The ordered sequence of numbers $(1, 2, \ldots, N)$ is modified as follows.

1. One of the numbers is removed. (We now have $N-1$ numbers in our ordered sequence.)

2. All remaining $N - 1$ numbers are permuted randomly. (We now have $N - 1$ numbers in our randomly-ordered sequence.)

For example, if $N = 5$, then we begin with $(1, 2, 3, 4, 5)$, can remove 2, resulting in $(1, 3, 4, 5)$, and a random permutation could be $(3, 4, 1, 5)$.

You are given N and the permuted sequence, and your goal is to identify the number that was removed. (In our example, the number 2 was removed.) Describe an efficient algorithm and provide its computational complexity. (There is no need to use pseudocode; describing in words is fine if your algorithm is simple.) More credit will be provided for algorithms with lower complexity.

Question 3 (Constants matter.)

The purpose of this question is to show that details of a computing system can impact what algorithms we prefer to use. Consider a computing system where memory accesses to the cache are 20 times faster than random access memory (RAM) access. Now consider the memory resources that two algorithms require for processing an input of size N.

- Algorithm 1 requires $f_{c1}(N) = 10N$ cache accesses and $f_{R1}(N) = \sqrt{N}$ RAM accesses.

- Algorithm 2 requires $f_{c2}(N) = 3N$ cache accesses and $f_{R2}(N) = N$ RAM accesses.

Algorithm 1 uses more cache accesses than Algorithm 2, but fewer RAM accesses.

(a) Because RAM is 20 times slower than cache, we can interpret RAM accesses as analogous to 20 cache accesses. Compute $f_1(N) = f_{c1}(N) + 20f_{R1}(N)$ and $f_2(N) = f_{c2}(N) + 20f_{R2}(N)$.

(b) For small values of N, $f_2(N)$ is faster (smaller); for large values of N, $f_1(N)$ is faster. Compute N^* for which $f_1(N^*) = f_2(N^*)$. You need not compute an actual number for N^*; an expression such as $N^* + 100/\sqrt{N^*} = \log_2(N^*)$ is fine.

(c) This question assumes that the cache is 20 times faster than RAM. What if it the cache is 10 times faster than RAM, 40 times faster? Discuss (there is no need for detailed calculations) whether the break-even point, N^*, will become larger or smaller.