Seat:

CompSci 260P Exam 2 Diagnostic Exam 1 Spring 2023

DO NOT OPEN EXAM UNTIL INSTRUCTED TO DO SO

SILENCE AND STORE ALL ELECTRONICS

This is a diagnostic exam intended to help you evaluate your readiness for the real exam.

The following rules apply to you, whether you think they do or not. Read and understand them; failure to abide by these rules, or directions given by course staff during the exam, may result in disciplinary action, including but not limited to a failing grade in the class.

- This exam is solely for students enrolled in this lecture. Anyone not enrolled in this lecture may not take an exam.
- You may not open the exam or begin writing until the instructor has explicitly given you permission to do so.
- Keep your UCI ID readily accessible during the test. Proctors may request to see it.
- This exam is closed book, closed notes, and is individual effort. Once course staff begin passing out exams, you may not communicate with anyone other than proctors for any reason, nor may you have electronics, including calculators watches and phones, available to you during the test for any reason. **YOU DO NOT NEED A CALCULATOR!**
- If you leave your seat during the test for any reason, your instructor may collect it and deem you to have turned it in. Do not ask proctors for an exemption to this, they are not authorized to grant such.
- If you are still seated at 7:35, you may not leave your seat until explicitly dismissed by the instructor. Leaving after 7:35 and before being dismissed may result in a grade penalty.
- If you believe a question is ambiguous, write at least two reasonable interpretations and indicate clearly which one you will be using. Then answer your question with that assumption. Unless your interpretation makes the problem much more trivial than intended, we will grade your response as if one of us had made that clarification.
- The purpose of the real exam is to evaluate how well *you* understand the material presented in the course. It is an academic integrity violation to do anything that subverts the goals of this assessment including, but not limited to, not doing your own work or submitting that of anyone else.
- We will only grade responses marked in the space provided for each question.

Nothing you write on this page will be graded. The next page in this booklet contains a spot to answer these questions. You may use this page as scratch paper if you would like, and room to do so exists.

Suppose we have an even number of people who are going to play (doubles) tennis, forming teams of size two. Each player has a *tennis rating*, (a positive number, where a higher number can be interpreted to mean a better player). The *quality* of each team is the *lower* tennis rating of a member of the team.

Our goal is to maximize the *sum* of the quality of teams formed.

Prove the following claim. The standard for how formal the proof needs to be is the same as it was in the relevant lecture and the homework

Consider the following algorithm. We pair the two best players together, then the third and fourth best players, and so on. This will maximize the sum of the quality of teams formed.

Claim: There is an optimal algorithm for this problem that makes the same decision I described in the previous paragraph.

Hint: if the two best aren't paired together, who are they paired with?

This question counts towards the Greedy portion of the Competency of Core Topics in your grade.

CompSci 260P- S23- Exam 2 Diagnostic 1

Write your answer to question 1 on this page

Nothing you write on this page will be graded. The next page in this booklet contains a spot to answer these questions. You may use this page as scratch paper if you would like, and room to do so exists.

Suppose I have a vector A of n distinct non-negative k-bit integers, where $n = 2^k - 1$ and k is a positive integer. Because of this setup, we know that exactly one integer in the range $[0, n]$ is missing. However, our fundamental operation is `bit[`i, j`]`, which returns the jth bit of $A[i]$. Using $\mathcal{O}(n)$ calls to `bit`, determine which integer is missing. Solutions with running time $\Omega(nk) = \Omega(n \log n)$ will earn zero credit. Note that accessing every bit will incur this running time; that is both a constraint and a hint.

Express the number of calls to `bit` that your algorithm makes, both as a recurrence relation and in closed form.

This question counts towards the Divide and Conquer portion of the Competency of Core Topics in your grade.

Write your answer to question 2 on this page