Algorithm Design

1. **DESCRIPTION:** Teams will demonstrate theoretical and practical knowledge of programming, data structures, and algorithms.

   A TEAM OF UP TO: 2

   EYE PROTECTION: NONE

   IMPOUND: NO

   APPROXIMATE TIME/EVENT TIME: 50 minutes

2. **EVENT PARAMETERS:** Teams will complete a written test and an on-site series of Coding Challenges in Python 3. The supervisor must run submitted code on the latest version of Python.
   - **a.** Each team must bring writing utensils and may bring a single calculator (which cannot be capable of internet access) and a single double-sided sheet of paper (8.5" x 11") containing notes of any kind.
   - **b.** Internet access is forbidden at all times. Violation of this rule will result in immediate disqualification.
   - **c.** The event supervisor will arrange to provide teams with a Python development environment (at minimum, a text editor and an installation of Python 3). A Python IDE such as PyCharm is recommended.
3. **THE COMPETITION:**
   - **a. Written Test.** Supervisors must write the test such that it can be completed without calculations which would require use of the laptop. The test will assess student knowledge in the following topics:
     - **i.** Python 3 syntax and programming knowledge. Includes but not limited to: variables, data types, control flow, methods/functions, classes, object-oriented programming, basic data structures.
     - **ii.** Time and space complexity. Includes but not limited to: Big O analysis, efficiency of various data structure operations, algorithm analysis.
   - **b. Coding Challenges.** Approximately 3-5 coding challenges.
     - **i.** The Coding Challenges should be completed in Python only. The only permitted external libraries are `math` and `numpy`.
     - **ii.** A Coding Challenge is a programming problem, and a solution is a Python program which: 1) reads an input file with input data, 2) parses and processes the input data appropriately, and 3) prints the output to standard output, in the format specified by the problem.
     - **iii.** Each Coding Challenge should be well-specified. For each challenge, supervisors will provide teams a set of sample input files and their respective expected outputs. This set should be representative of the test cases that will be used for scoring.
     - **iv.** Supervisors should write multiple test cases to evaluate the correctness of each team's solution. Good test cases will test edge cases. For some Challenges, supervisors may choose to limit the run time and use large input sizes to test the efficiency of the team's solution.
     - **v.** Each solution file must have a header comment containing the team name and team number. Teams should submit their source code according to supervisor instructions.
4. **SAMPLE CODING CHALLENGES**: Note that the following are underspecified – supervisors should provide more detailed explanations of the expectations for each Challenge.
   - **a.** Given a sequence of DNA, determine the relative frequency of each nucleotide
   - **b.** Given height vs. time data for an oscillating spring, determine the period of oscillation.
   - **c.** Given a large chemical equation, determine the smallest integer coefficients
   - **d.** Given a phylogenetic tree, determine the two species which are furthest from each other
5. **SCORING:**
   - **a.** High score wins. The written test and coding component are weighted equally.
   - **b.** The coding component subscore is split: 80% for correctness (passing test cases) and 20% points for code organization and good programming practices (useful inline documentation, following pythonic style, etc).

c. Supervisors must indicate how much each test question is worth, and how much each Coding Challenge is worth. If not specified, each test question is weighted equally and each Coding Challenge is weighted equally.

d. Tiebreaker questions are designated by the supervisors.

6. **RECOMMENDED RESOURCES:**
   - **a.** http://codingbat.com/python
     - **i.** CodingBat can help students develop basic programming skills in python. No Python installation required.
   - **b.** https://leetcode.com
     - **i.** Intermediate to challenging programming problems