Assignment 00

Algorithms for Sequence Analysis

Sven Rahmann and Jens Zentgraf

16.04.2023

Tutorials and Assignments

- Assignments will be given out each Thursday (this one: Tuesday)
- Approximately 12 assignments with 4 tasks each, 4 points per task
- You generally have one week to complete and submit an assignment.
- Assignments include **theory** and **programming** tasks.
- To qualify for the final exam, you need:
    - at least 33% of all points of assignment 00,
    - at least 50% of all points from theory tasks,
    - at least 50% of all points from programming tasks, **and**
    - present at least 2 solutions in the tutorial.
- You can work in **groups of two students** (but not more!)

How to Hand in your Solutions

Teams

- Use the tutorials or the forum to form teams
- It is enough if one person uploads for the team.
- For each task make clear who worked on this task.
    - Write your name next to the task if you worked on it.
    - You only get points for a task if you work on it (and your name is at the task)!

CMS

- You submit your solutions via the cms.
- Separate submissions for theory and programming tasks.

Programming tasks

- If necessary, a code framework and tests are provided by us in Python.

Plagiarism

- **Never** plagiarize or copy solutions!
    - Do **not** use code/explanations from the internet!
    - Do **not** copy code/explanations from other students!
    - Do **not** copy old solutions form the previous semesters!
    - If you plagiarize you are **not allowed** to take the exam!
    - If you plagiarize you can be **reported** to the examination board!
- If you work in a team:
    - For each task make clear who worked on this task.
    - Write your name next to the task if you worked on it.
    - You only get points for a task if you work on it (and your name is at the task)!
    - If your name is at a task and it is a plagiarism, you are responsible!
      Check your solution before it is uploaded.

Assignment 00 (3 points)

Implement the following algorithm for a given positive integer x.

Given a positive integer x, obtain a new value for x by the following rules:
1. If x is even, we divide x by 2.
2. If x is odd, we multiply x by 3, and add 1.

$$x \leftarrow \begin{cases} x/2 & \text{if } x \text{ is even,} \\ 3 \cdot x + 1 & \text{if } x \text{ is odd.} \end{cases}$$

This is iterated until x reaches 1. If $x = 1$, it enters a cycle $1 \rightarrow 4 \rightarrow 2 \rightarrow 1$.

Extra credit

Find the smallest positive integer x, for which this never happens.

Algorithmic Bioinformatics